

2017 Machine Learning with R

Clustering

강필성

고려대학교 산업경영공학부

pilsung_kang@korea.ac.kr

목차

I

군집화 소개

II

K-평균 군집화: K-Means Clustering

III

계층적 군집화: Hierarchical Clustering

IV

자기 조직화 지도

V

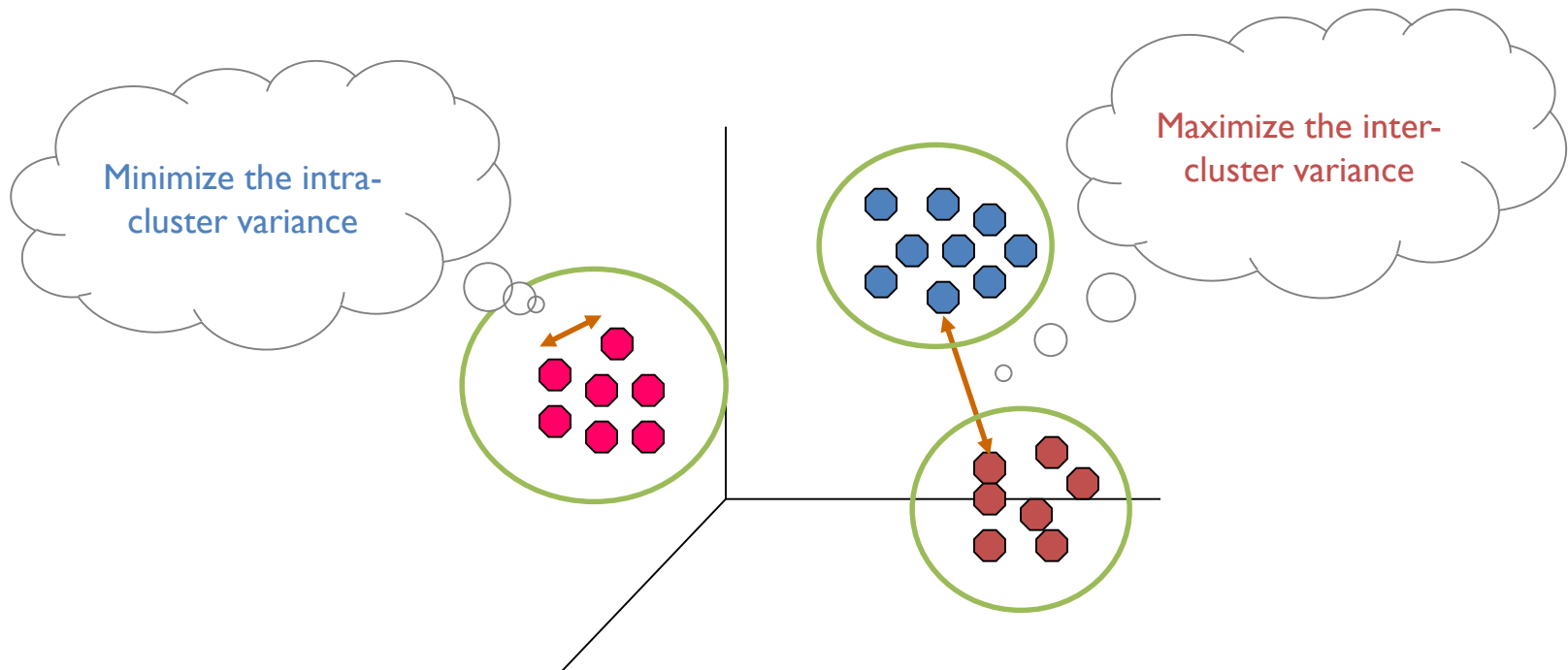
R 실습

군집화: Clustering

❖ 군집화(Clustering)

■ 관측치들의 집단을 판별

- ✓ 동일한 집단에 소속된 관측치들은 서로 유사할수록 좋음
- ✓ 상이한 집단에 소속된 관측치들은 서로 다를수록 좋음

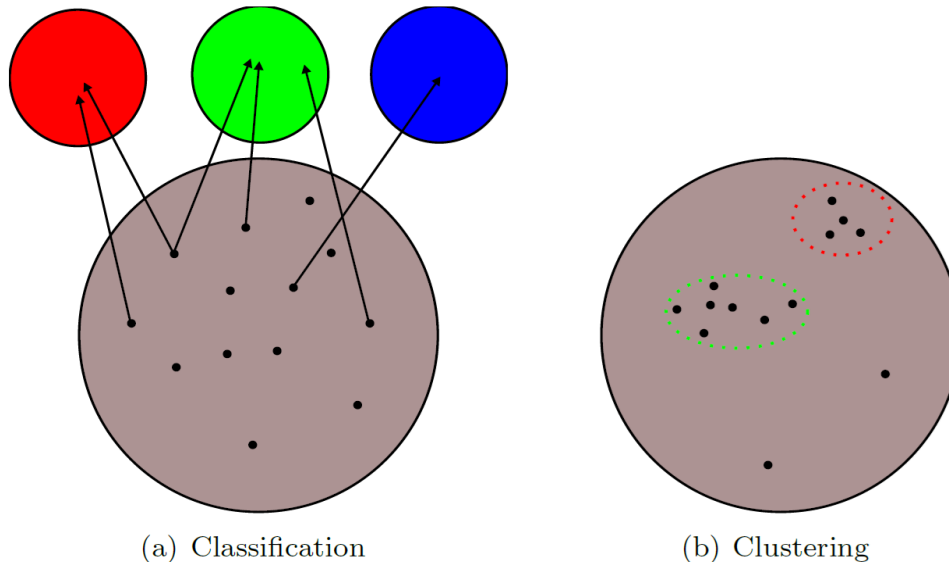


군집화: Clustering

Andrews and Fox (2007)

❖ 분류 (Classification) vs. 군집화(Clustering)

- **분류(Classification)**: 범주의 수 및 각 개체의 범주 정보를 사전에 알 수 있으며, 개체의 입력 변수 값들로부터 범주 정보를 유추하여 새로운 개체에 대해 가장 적합한 범주로 할당하는 문제 (supervised learning)
- **군집화(Clustering)**: 군집의 수, 속성, 멤버십 등이 사전에 알려져 있지 않으며 최적의 구분을 찾아가는 문제 (unsupervised learning)



군집화: 적용사례

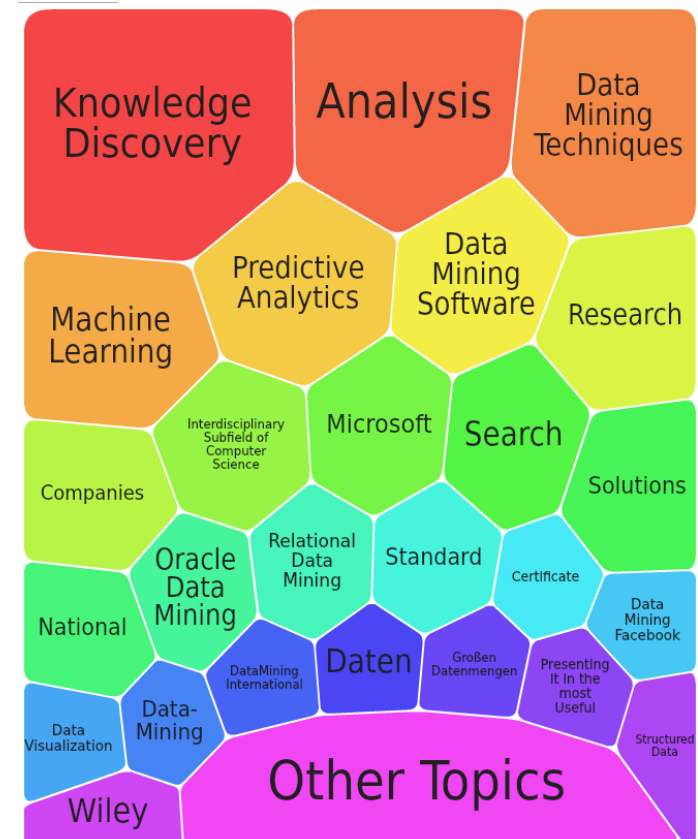
❖ 군집화 적용 사례

■ 데이터에 대한 이해

- ✓ 웹브라우징 시 유사한 문서들을 표시
- ✓ 유사한 기능을 수행하는 유전자/단백질 집합
- ✓ 유사한 추세를 나타내는 주식 종목들 등

Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

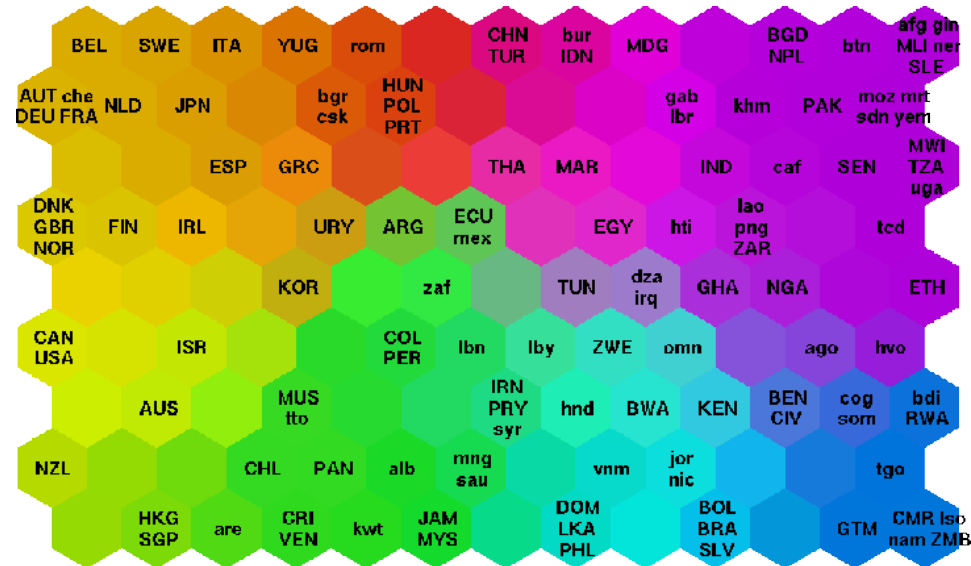
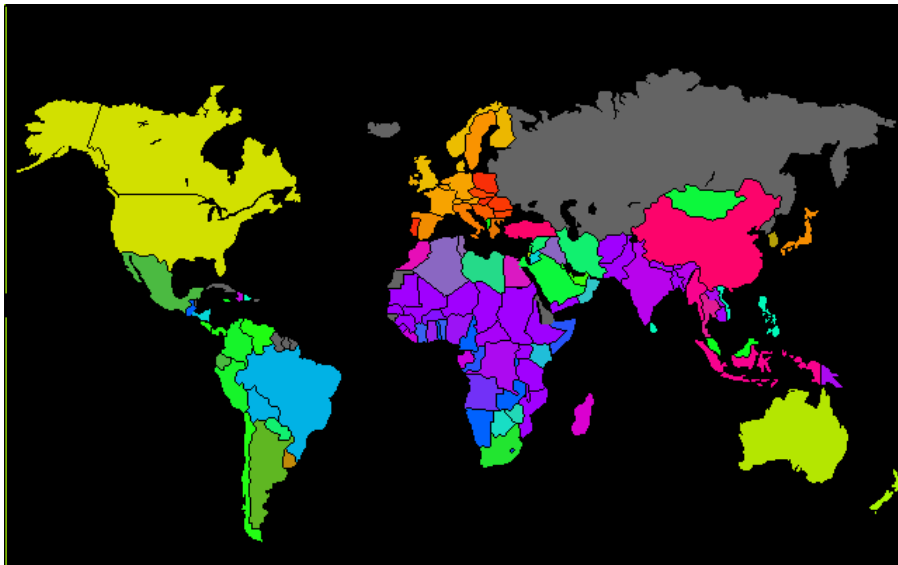
Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%), organizations (43%) ● Ahavat Israel - The Amazing Jewish Website! ● Israel and Judaism ● Judaica Collection
2 View Results Refine Query Based On This Cluster	15	Ministry of Foreign Affairs (33%), Ministry (87%) ● Publications and Data of the BANK OF ISRAEL ● Consulate General of Israel to the Mid-Atlantic Region ● The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%) ● Interactive Israel tourism guide - Jerusalem ● Ambassade d'Israel ● Travel to Israel Opportunities
4 View Results Refine Query Based On This Cluster	7	Middle East (57%), History (57%); WAR (42%), Region (42%), Complete (42%), Listing (42%), country (42%) ● Israel at Fifty: Our Introduction to The Six Day War ● Machal - Volunteers in the Israel's War of Independence ● HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	Economy (68%), Companies (55%), Travel (55%) ● Israel Hotel Association ● Israel Association of Electronics Industries ● Focus Capital Group - Israel



군집화: 적용사례

❖ 군집화 적용 사례

- 대량의 데이터에 대한 요약
 - ✓ 고차원의 데이터를 저차원으로 축약하여 정보를 요약
- 시각화(Visualization)과 밀접한 관계가 있음

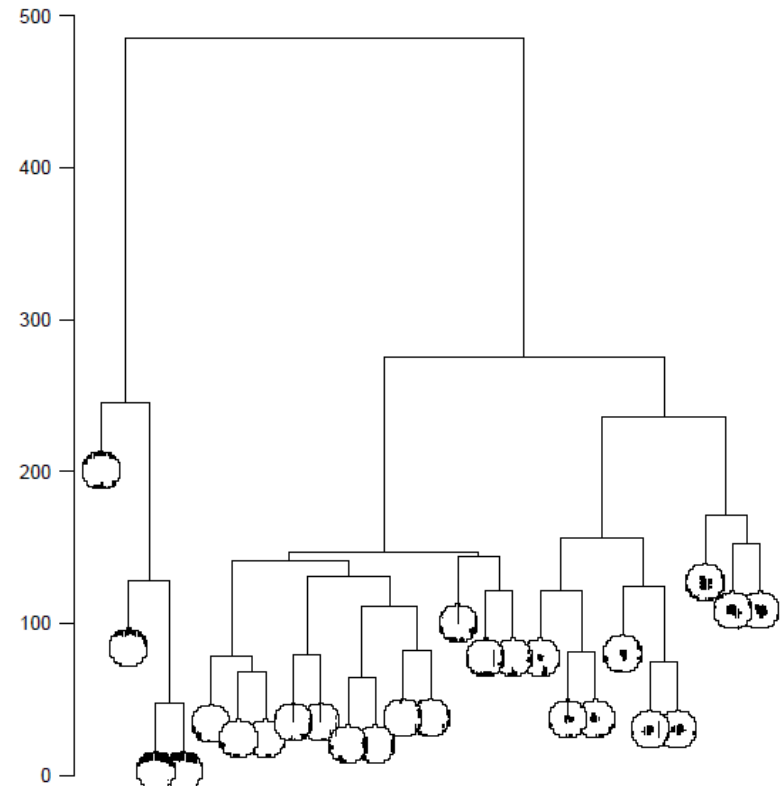
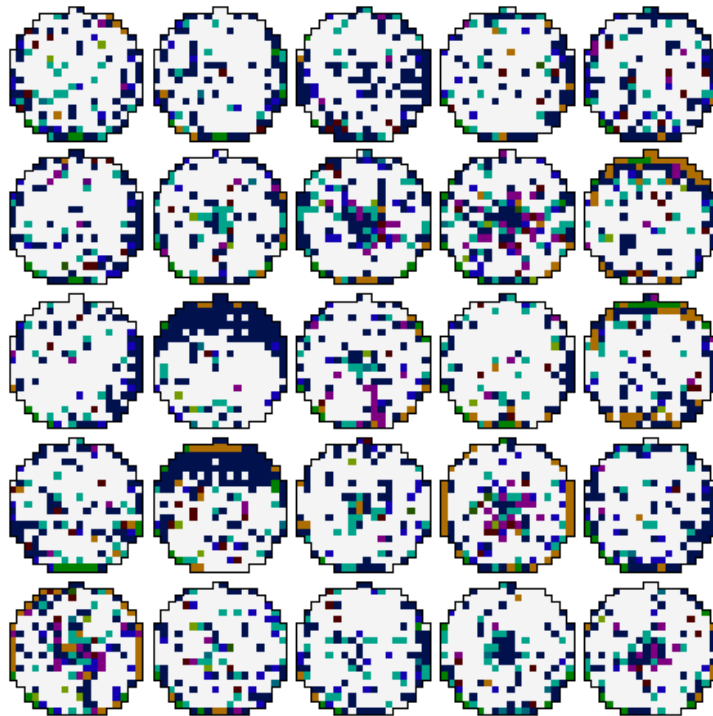


군집화: 적용사례

❖ 군집화 적용 사례

■ 웨이퍼 Fail bit map 군집화

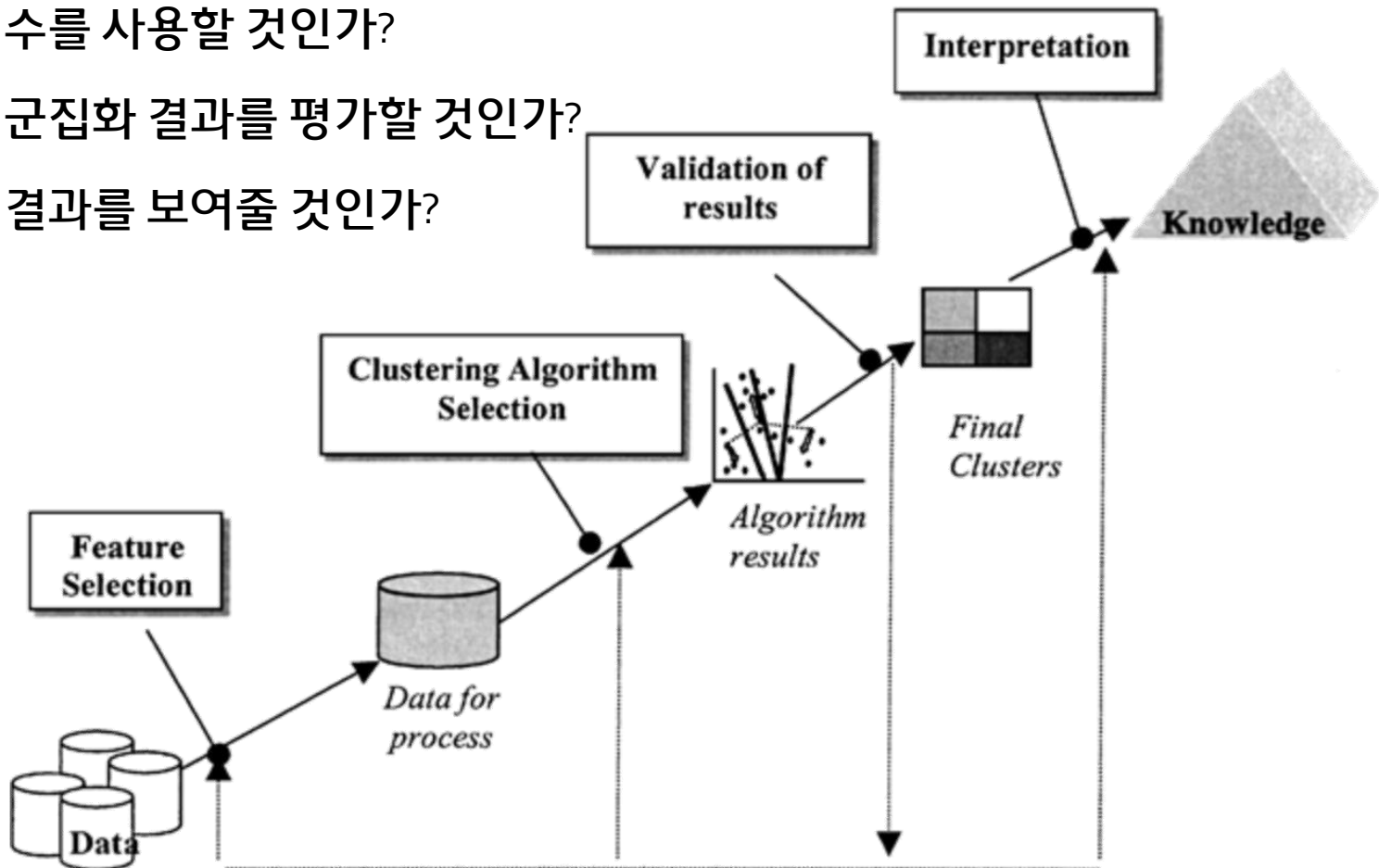
Sample lot exhibiting spatial patterning



군집화: 수행 절차

❖ 일반적인 군집화 수행 절차

- 어떤 변수를 사용할 것인가?
- 어떻게 군집화 결과를 평가할 것인가?
- 어떻게 결과를 보여줄 것인가?



군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

▪ 최적의 군집 수 결정



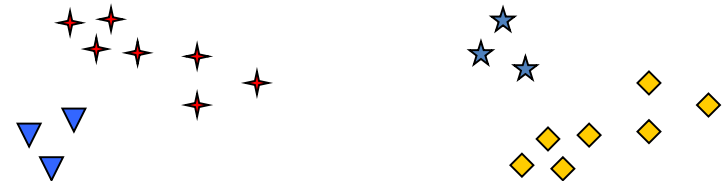
How many clusters?



Six Clusters



Two Clusters



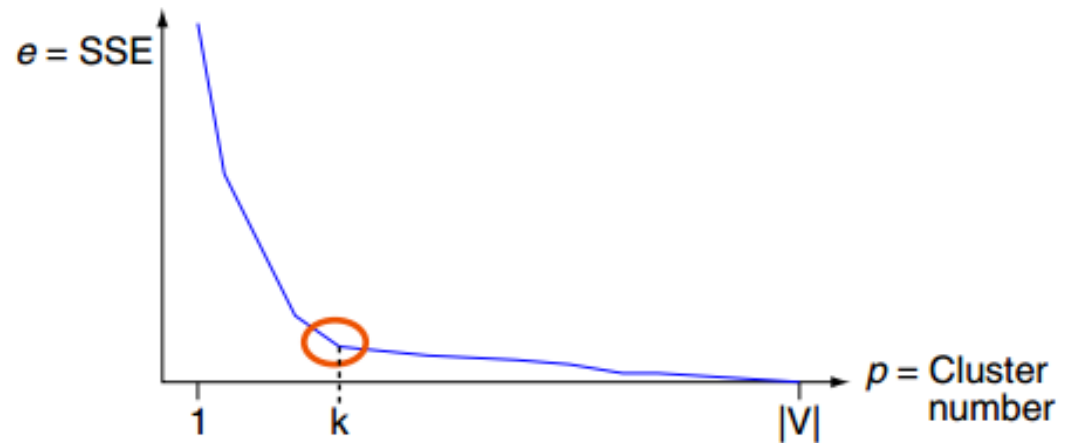
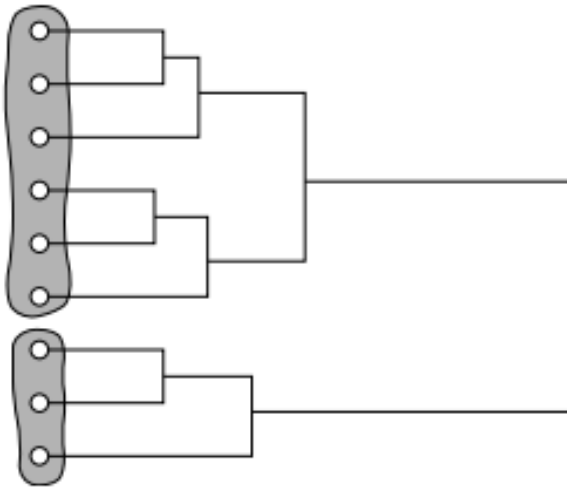
Four Clusters

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

■ 최적의 군집 수 결정

- ✓ 다양한 군집 수에 대해 성능 평가 지표를 도출하여 최적의 군집 수 선택
- ✓ Elbow point에서 최적 군집 수가 결정되는 경우가 일반적임



군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 군집화 결과를 어떻게 평가할 것인가?
- 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure 부재

❖ 군집화 평가 지표는 다음과 같이 세 가지 카테고리로 구분할 수 있음

- External: 정답 레이블과의 비교를 통해 성능 평가 (현실적으로 불가능)
- Internal: “군집이 얼마나 컴팩트한가”에 보다 초점을 둠
- Relative: “군집이 얼마나 컴팩트한가”와 “군집끼리 얼마나 다른가”를 동시에 고려하고자 함

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 군집화 결과를 어떻게 평가할 것인가?
- 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure

부재

External

Internal

Relative



- ☐ Rand Statistic
- ☐ Jaccard Coefficient
- ☐ Folks and Mallows index
- ☐ (Normalized) Hubert Γ statistic

- ☐ Cophenetic Correlation Coefficient
- ☐ Sum of Squared error (SSE)
- ☐ Cohesion and separation

- ☐ Dunn family of indices
- ☐ Davies-Bouldin (DB) index
- ☐ Semi-partial R-squared
- ☐ SD validity index
- ☐ Silhouette

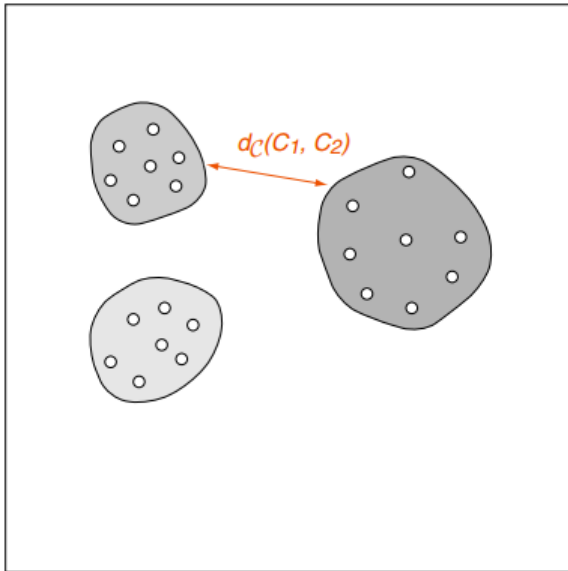
군집화: 고려 사항

❖ 군집화 평가를 위해 필요한 세 가지 지표 정의

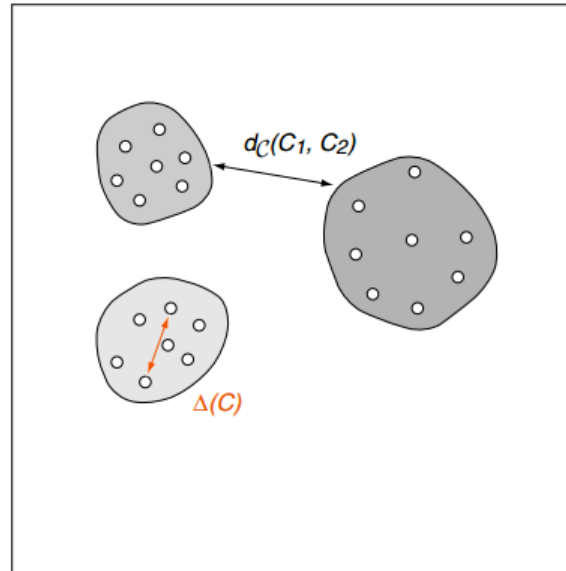
■ 군집화가 잘 되어 있다면

✓ (1)번 지표의 값은 크고 (2)번과 (3)번 지표의 값은 상대적으로 작게 나타날 것임

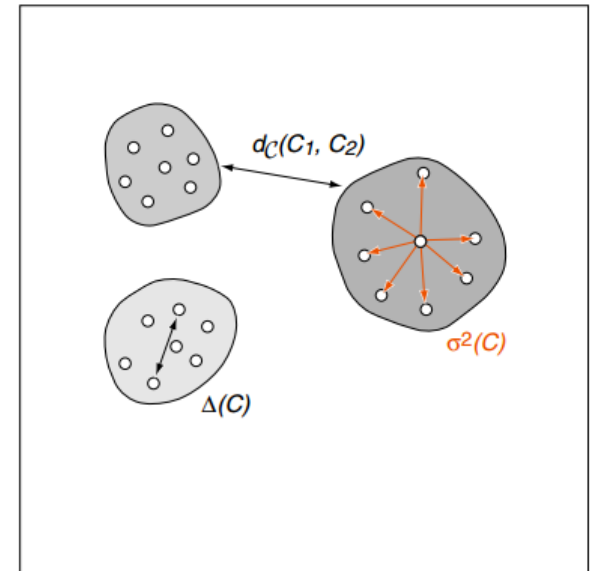
(1) Distance between two clusters



(2) Diameter of a cluster



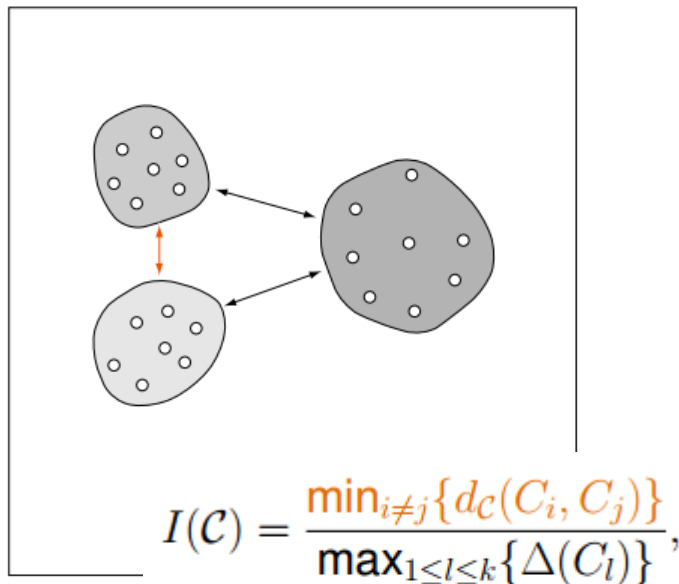
(3) Scatter within a cluster (SSE)



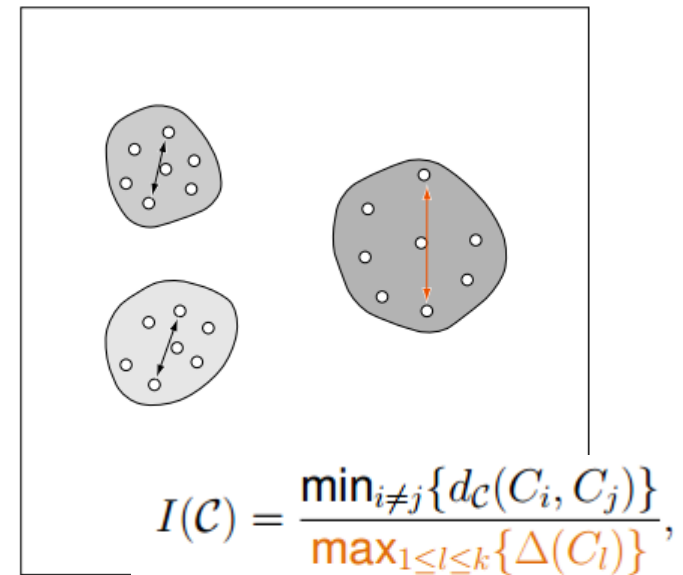
군집화: 고려 사항

❖ 군집화 평가 지표 1: Dunn Index

- Dunn Index는 군집 내 거리((1)번 지표)중 가장 작은 값을 분자로, 군집의 지름((2)번 지표) 중 가장 큰 값을 분모로 정의함
- Dunn Index는 클수록 우수한 군집화 결과라고 할 수 있음



$I(C) \rightarrow \max$



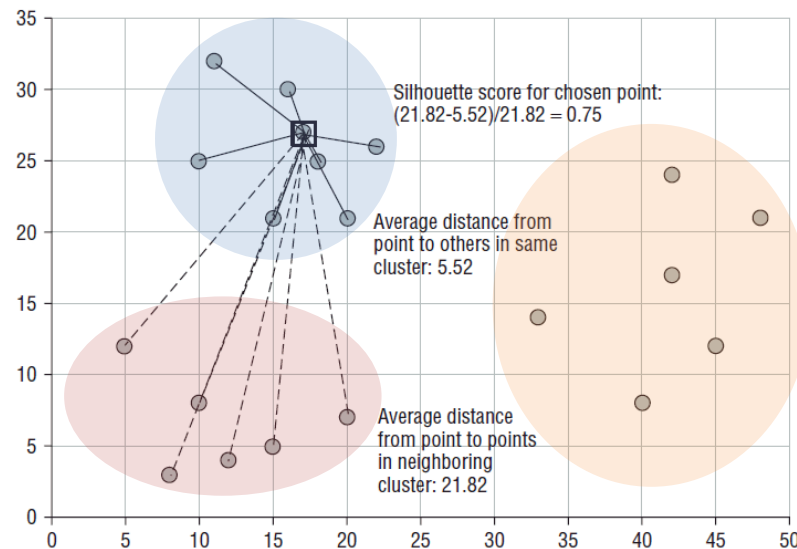
$I(C) \rightarrow \max$

군집화: 고려 사항

❖ 군집화 평가 지표 2: Silhouette

- $a(i)$: 개체 i 로부터 같은 군집 내에 있는 모든 다른 개체들 사이의 평균 거리
- $b(i)$: 개체 i 로부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 가장 작은 값

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases} \quad -1 \leq s(i) \leq 1$$

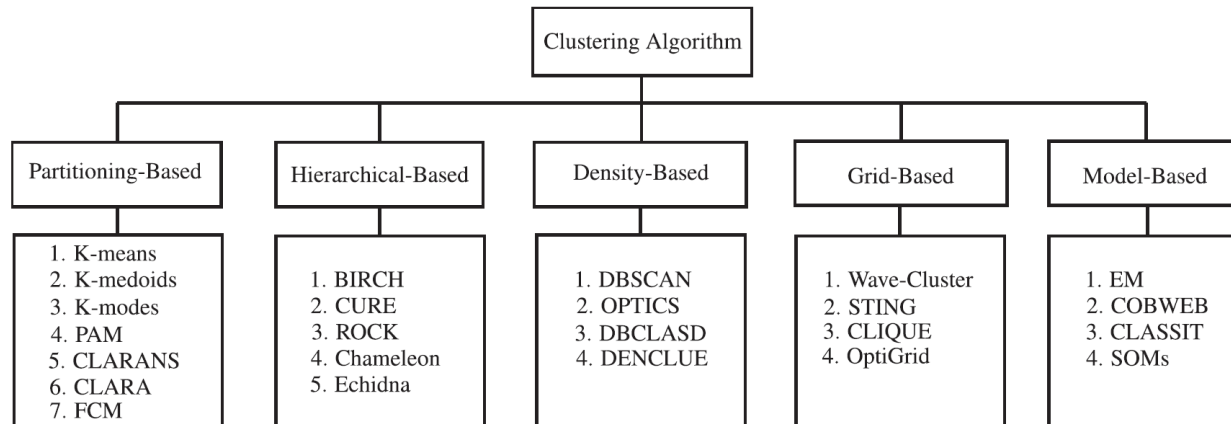


군집화 종류

❖ Hard Clustering vs. Soft Clustering

▪ Hard Clustering (Crisp Clustering)

- ✓ 서로 겹치지 않는(non-overlapping) 군집 생성
- ✓ 각 개체는 오직 하나의 군집으로만 할당됨



▪ Soft Clustering (Fuzzy Clustering)

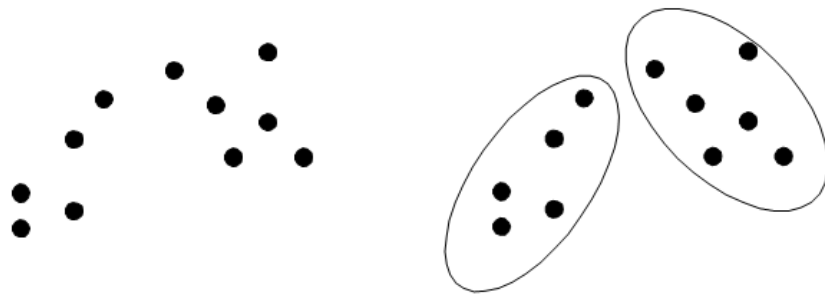
- ✓ 겹치는 군집을 생성하는 것도 가능함
- ✓ 한 개체는 여러 개의 군집에 확률적인 할당이 될 수 있음

군집화: 알고리즘

❖ 군집화 알고리즘의 종류

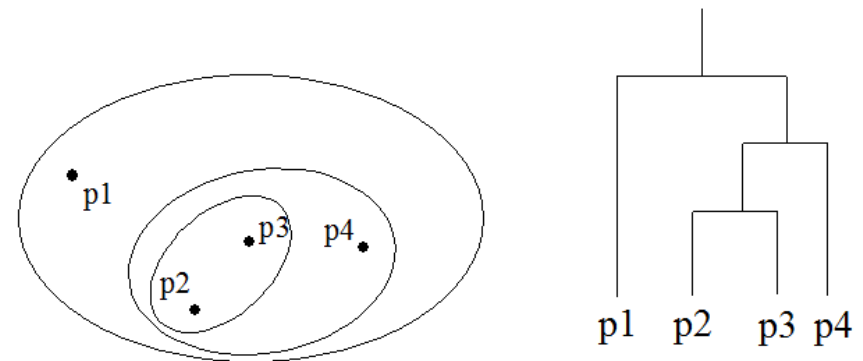
■ 분리형 군집화

- ✓ 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- ✓ 각 개체들은 사전에 정의된 군집 수 중 하나에 속하는 결과를 도출함



■ 계층적 군집화

- ✓ 개체들을 가까운 집단부터 차근차근 묶어가는 방식
- ✓ 군집화 결과 뿐만 아니라 유사한 개체들이 결합되는 절차(dendrogram)도 생성

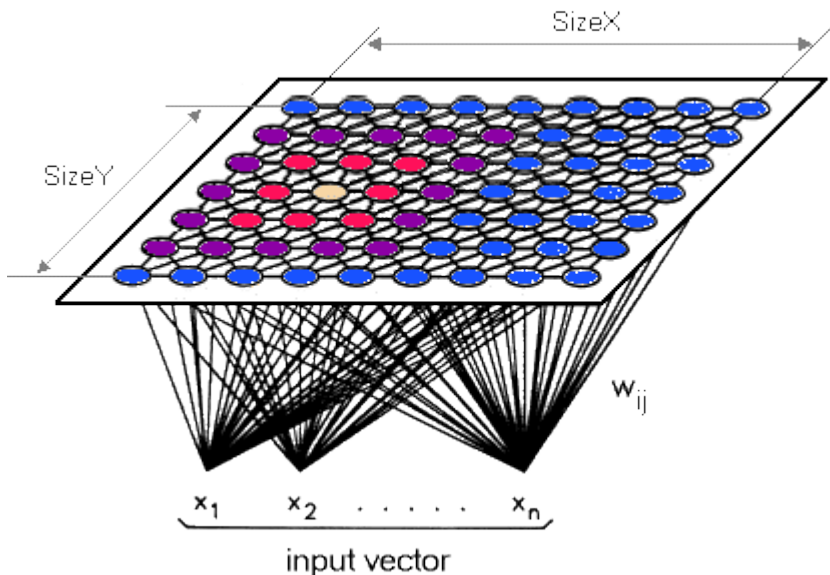


군집화: 알고리즘

❖ 군집화 알고리즘의 종류

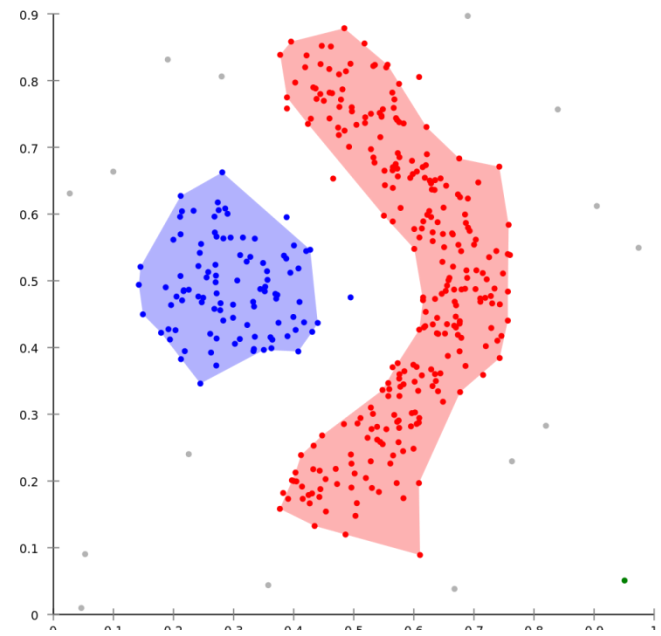
■ 자기조직화 지도: Self-Organizing Map (SOM)

- ✓ 2차원의 격자에 각 개체들이 대응하도록 인공신경망과 유사한 학습을 통해 군집 도출



■ 밀도 기반 군집화

- ✓ 데이터의 분포를 기반으로 높은 밀도를 갖는 세부 영역들로 전체 영역을 구분



목차

I

군집화 소개

II

K-평균 군집화: K-Means Clustering

III

계층적 군집화: Hierarchical Clustering

IV

자기 조직화 지도

V

R 실습

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화

■ 대표적인 분리형 군집화 알고리즘

- ✓ 각 군집은 하나의 중심(centroid)을 가짐
- ✓ 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 생성
- ✓ 사전에 군집의 수 K가 정해져야 알고리즘을 실행할 수 있음

$$\mathbf{X} = C_1 \cup C_2 \dots \cup C_K, \quad C_i \cap C_j = \phi, \quad i \neq j$$

$$\arg \min_{\mathbf{C}} \sum_{i=1}^K \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{c}_i\|^2$$

K-평균 군집화: K-Means Clustering

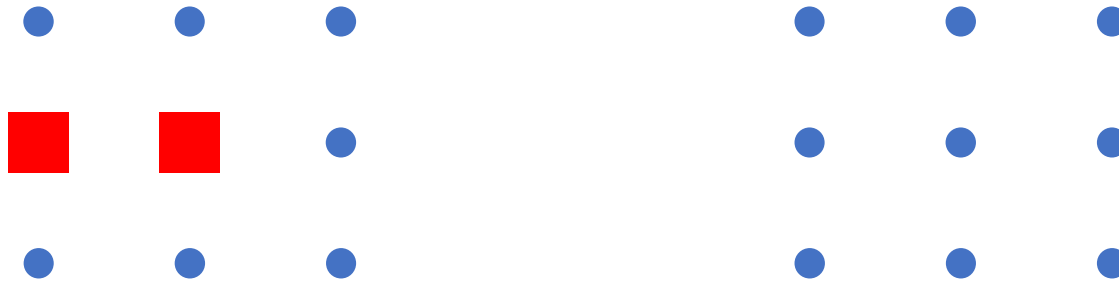
❖ K-평균 군집화 (K-Means Clustering) 수행 절차

- 1단계: K개의 초기 군집 중심(initial centroid) 설정
- 2단계: 다음 절차를 반복
 - ✓ 모든 개체를 가장 가까운 군집 중심에 할당하여 군집 구성
 - ✓ 할당된 개체들을 이용하여 군집 중심을 재설정
 - ✓ 종료 조건: 모든 군집 중심의 위치가 변하지 않고, 모든 개체의 군집 할당 결과에 변화가 없을 때 알고리즘 종료
- Note: 초기 중심은 종종 무작위로 선택되며 따라서 군집화의 결과가 초기 중심 설정에 따라 다르게 나타나는 경우가 발생할 수도 있음

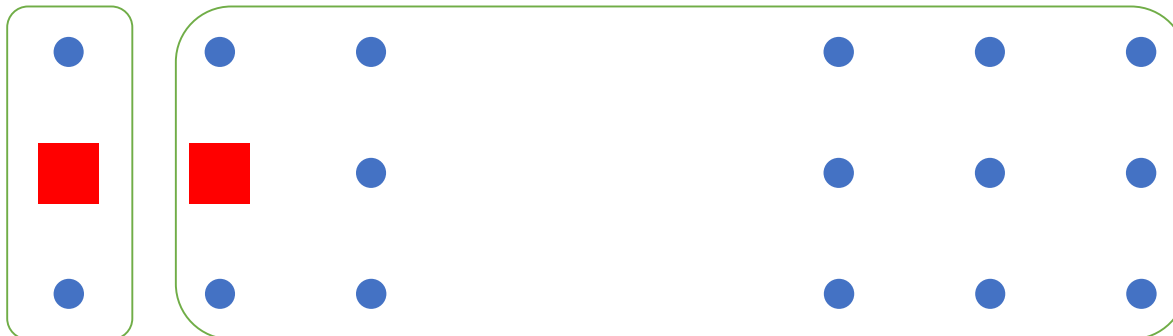
K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차 예시

- 1단계: K개의 초기 군집 중심(initial centroid) 설정



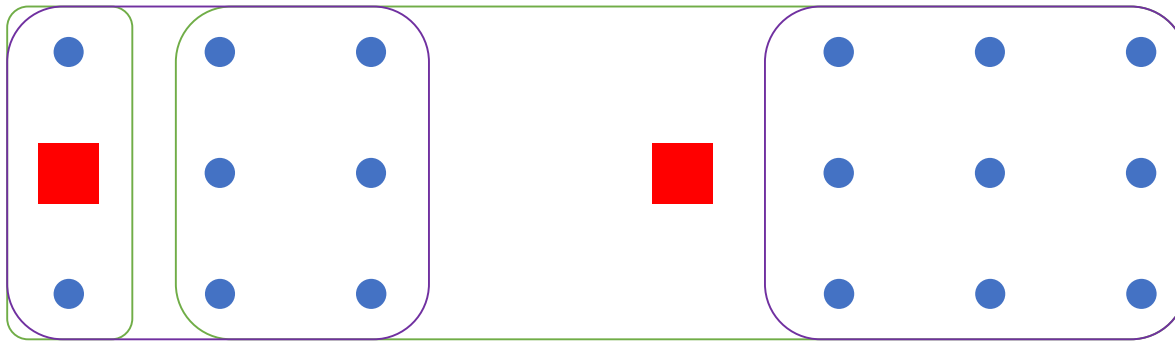
- 2-1단계(1회차): 모든 개체를 가장 가까운 중심에 할당
- 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정



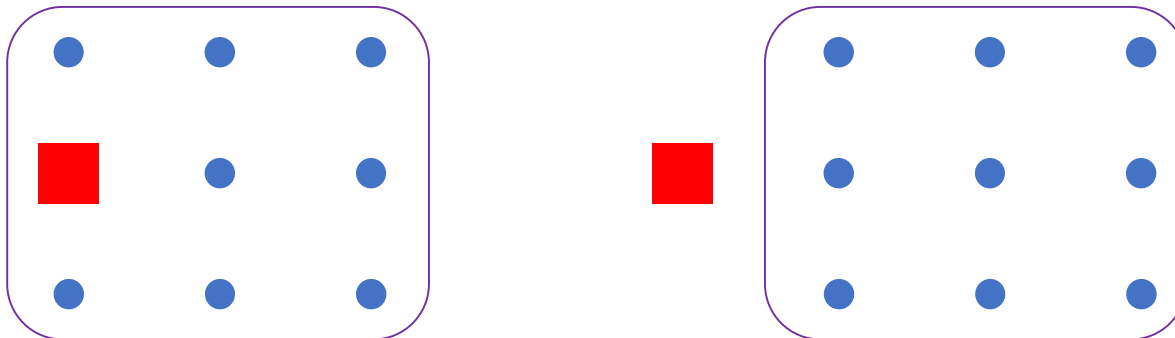
K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차 예시

- 2-1단계(2회차): 모든 개체를 가장 가까운 중심에 할당



- 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정

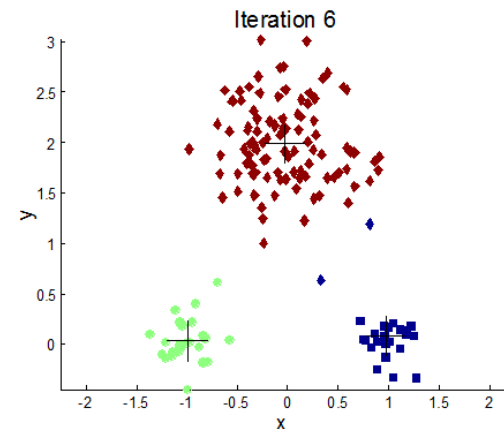
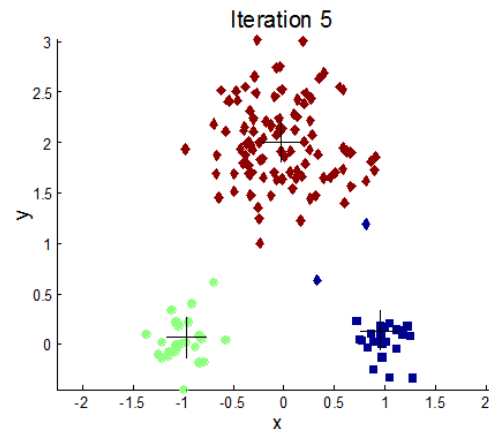
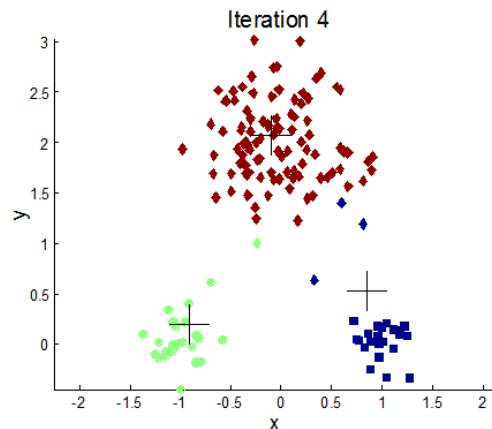
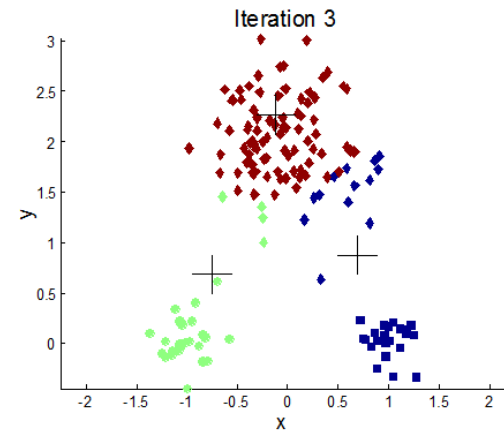
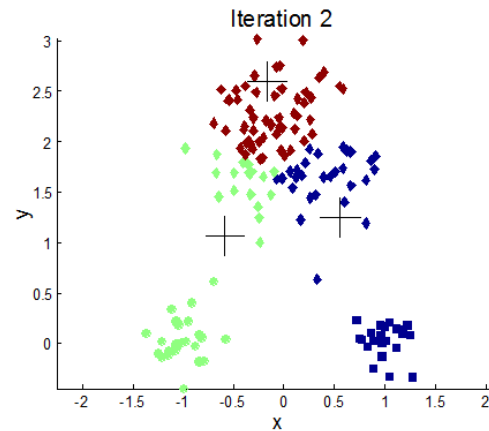
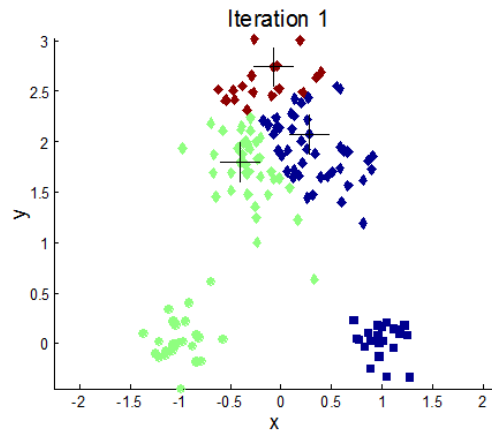


- 군집 중심과 개체 할당에 변화가 없으므로 알고리즘 종료

K-평균 군집화: K-Means Clustering

❖ 초기 중심 설정의 영향

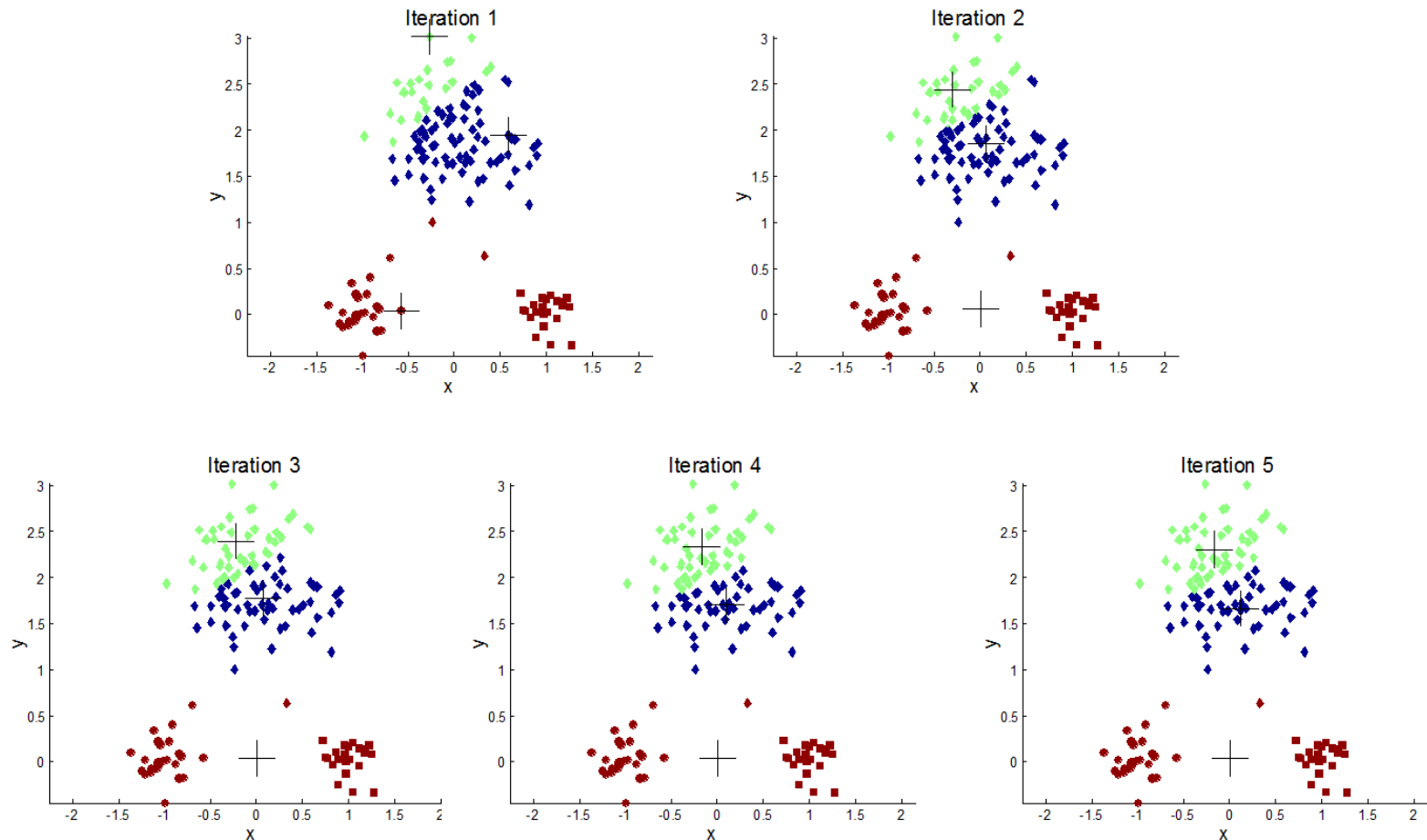
■ 바람직한 결과



K-평균 군집화: K-Means Clustering

❖ 초기 중심 설정의 영향

■ 바람직하지 않은 결과

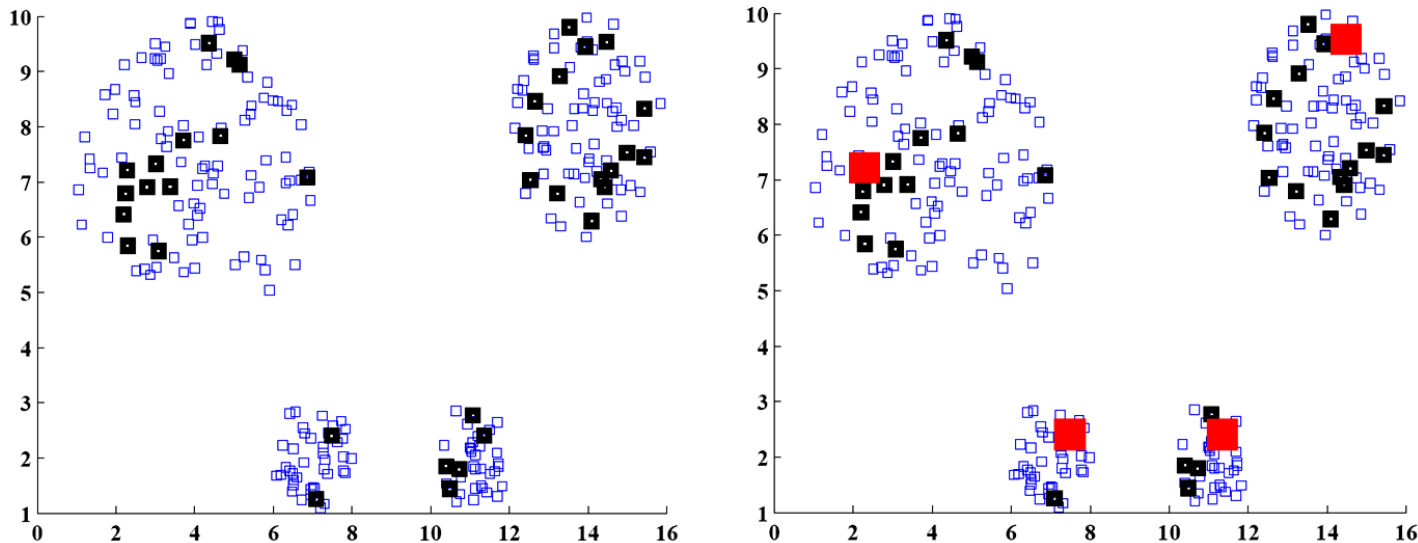


K-평균 군집화: K-Means Clustering

❖ 무작위 초기 중심 설정의 위험을 감소하고자 하는 다양한 연구가 존재

- 반복적으로 수행하여 가장 여러 번 나타나는 군집을 이용
- 전체 데이터 중 일부만 샘플링하여 계층적 군집화를 수행한 뒤 초기 군집 중심 설정

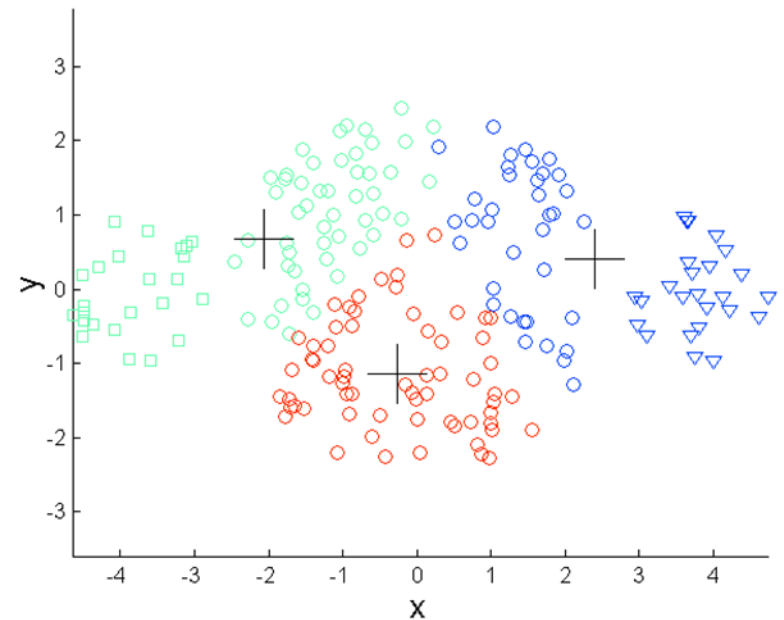
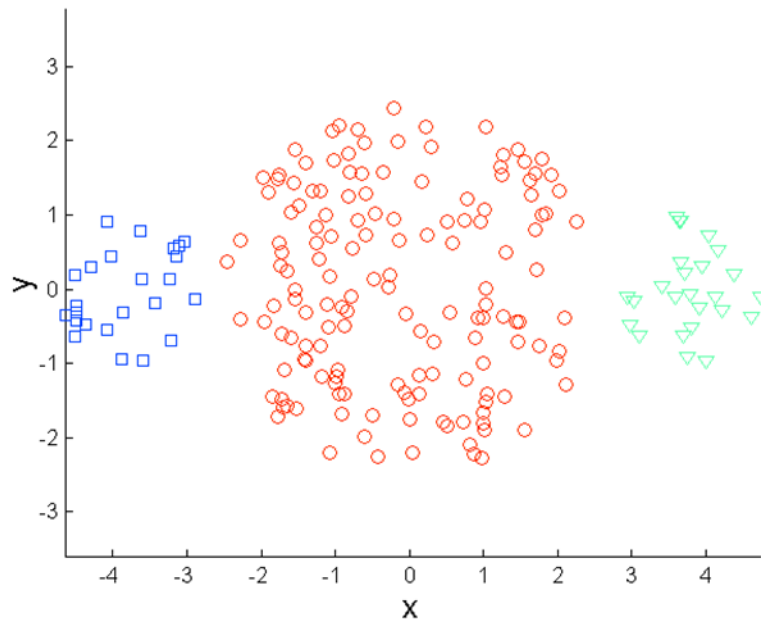
- 데이터 분포 $\mathcal{L}(\mathbf{x}_s | \mathbf{S}, \mathbf{C}) = d_G(\mathbf{x}_s, \mathbf{S}) \times \frac{1}{1 + \exp(-d_R(\mathbf{x}_s, \mathbf{S}))}$



K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

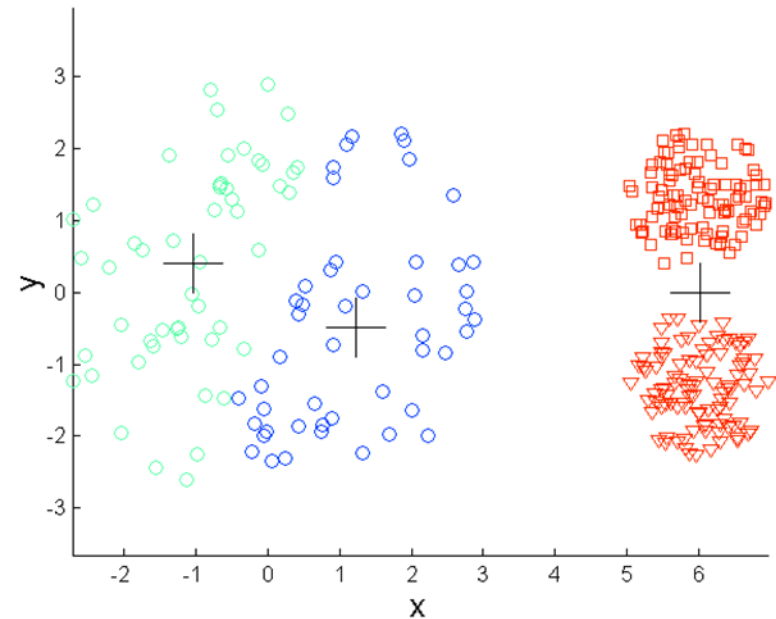
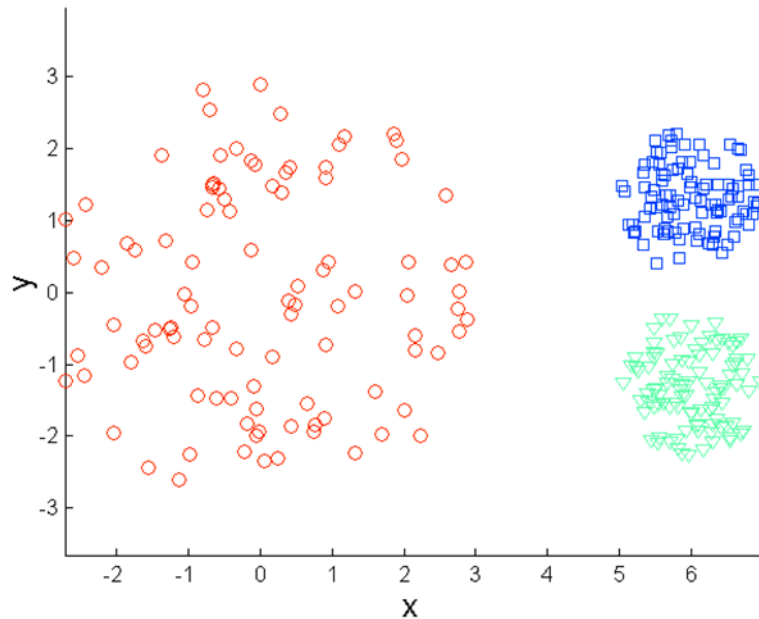
- 문제점 1: 서로 다른 크기의 군집을 잘 찾아내지 못함



K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

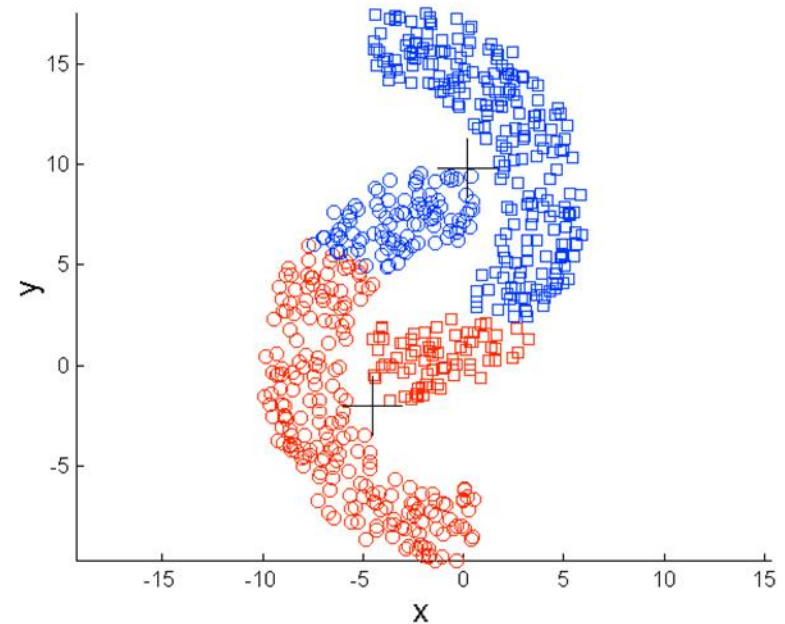
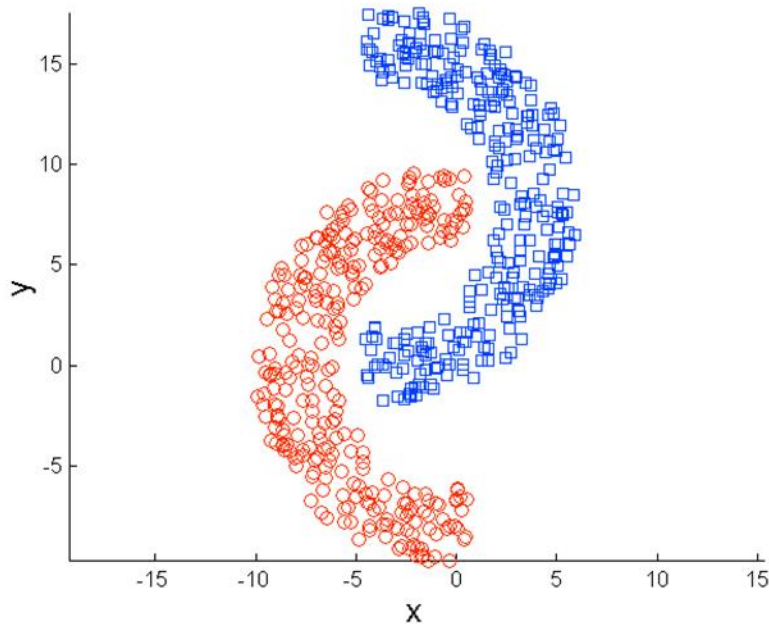
- 문제점 2: 서로 다른 밀도의 군집을 잘 찾아내지 못함



K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

- 문제점 3: **구형이 아닌 형태**의 군집을 판별하기 어려움



목차

I

군집화 소개

II

K-평균 군집화: K-Means Clustering

III

계층적 군집화: Hierarchical Clustering

IV

자기 조직화 지도

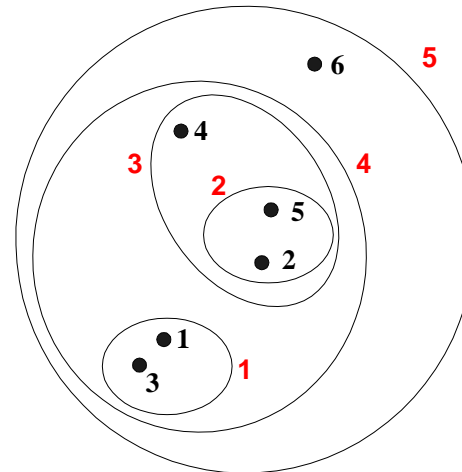
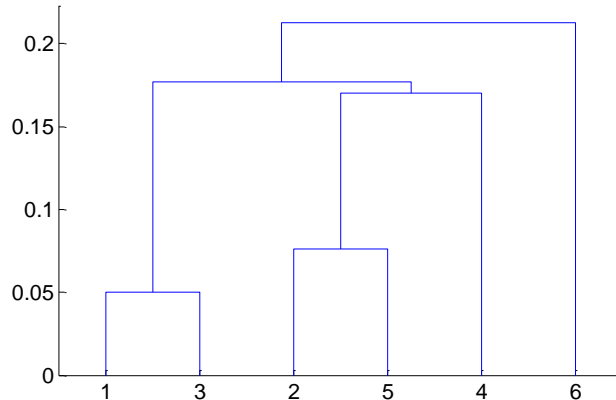
V

R 실습

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화

- 계층적 트리모형을 이용하여 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합
- 덴드로그램(Dendrogram)을 통해 시각화 가능
 - ✓ Dendrogram: 개체/군집들이 결합되는 순서를 나타내는 트리형태의 구조



계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화의 장점

- 사전에 군집의 수를 정하지 않아도 수행 가능
 - ✓ Dendrogram이 생성된 후 적절한 수준에서 자르면 그에 해당하는 군집화 결과 생성
- 특정 분야(domain)에서는 이 dendrogram이 유의미한 계통체계(taxonomies)를 표현하기도 함

❖ 계층적 군집화의 두 가지 방식

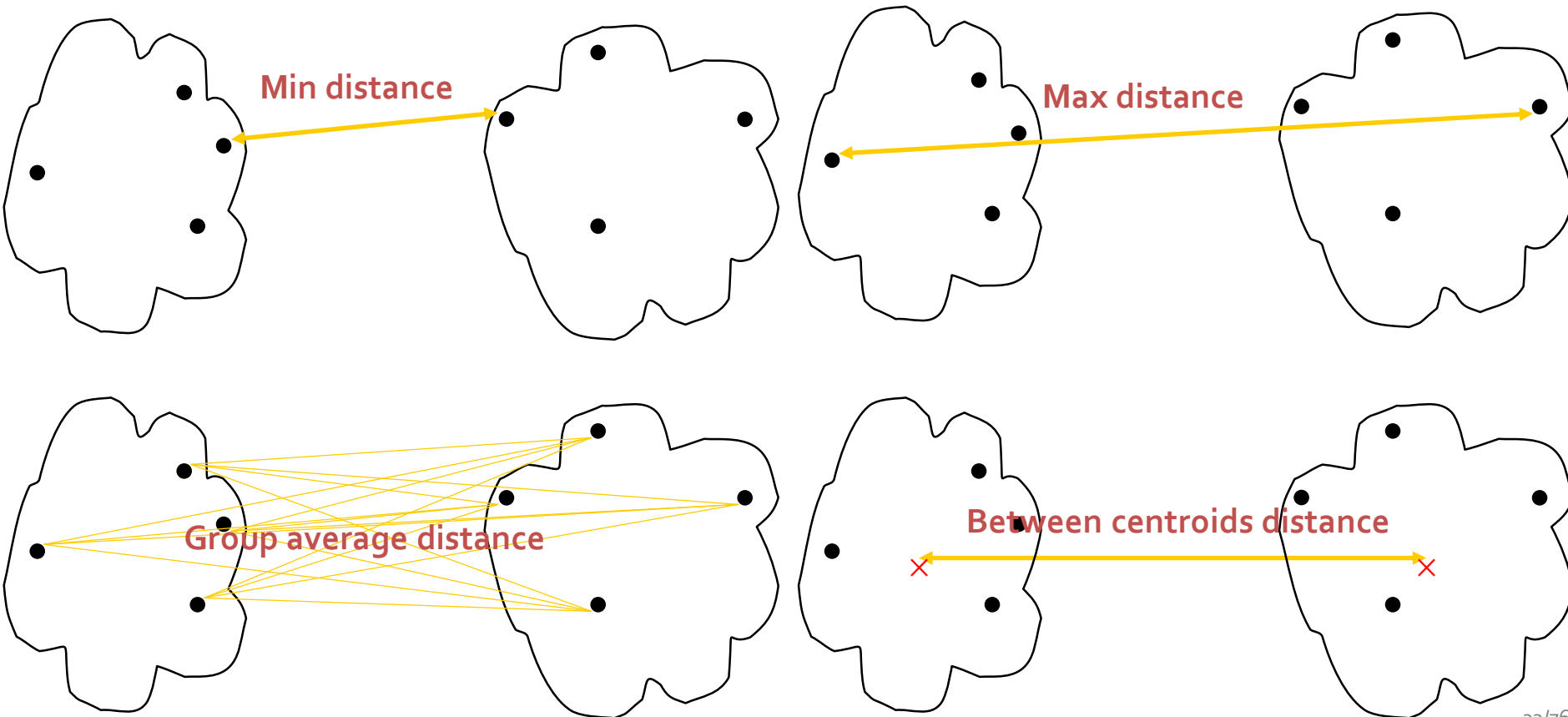
- **상향식 군집화: Agglomerative clustering**
 - ✓ 초기에 모든 개체들을 개별적인 군집으로 가정
 - ✓ 각 단계에서 유사한 개체/군집 결합 → 모든 개체들이 하나의 군집으로 통합되면 완료
- **하향식 군집화: Divisive clustering**
 - ✓ 모든 개체가 하나의 군집으로 이루어진 상태에서 출발
 - ✓ 각 단계에서 가장 유의미하게 구분되는 지점을 판별하여 지속적으로 데이터를 분할

계층적 군집화: Hierarchical Clustering

❖ 상향식 군집화 알고리즘

■ 핵심 수행 절차: 두 군집 사이의 유사도/거리 측정

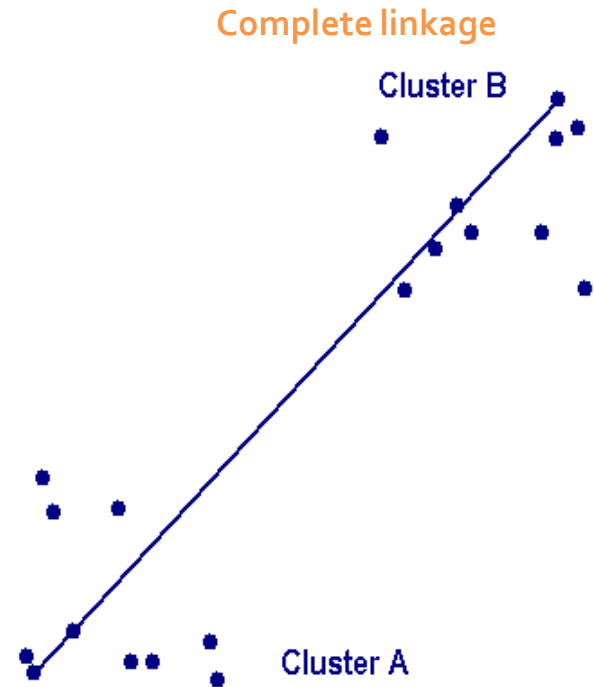
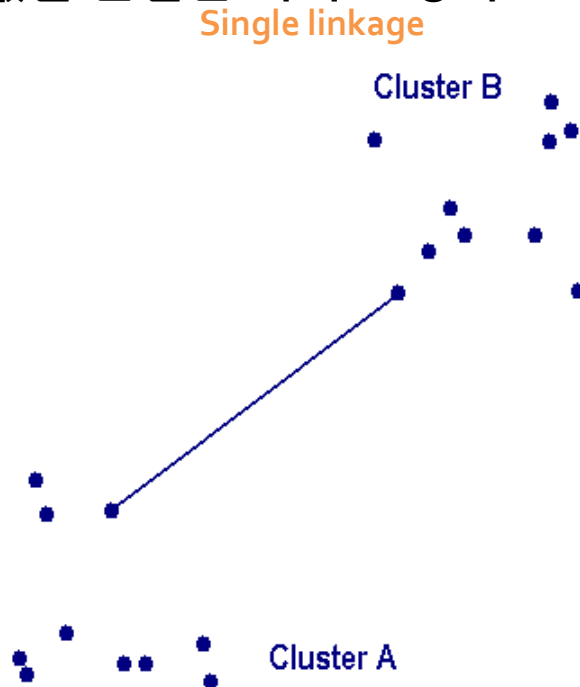
✓ Min, max, group average, between centroid, etc.



계층적 군집화: Hierarchical Clustering

❖ 상향식 군집화 알고리즘

- Single linkage (minimum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 가까운 값을 군집간 거리로 정의
- Complete linkage (maximum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 먼 값을 군집간 거리로 정의

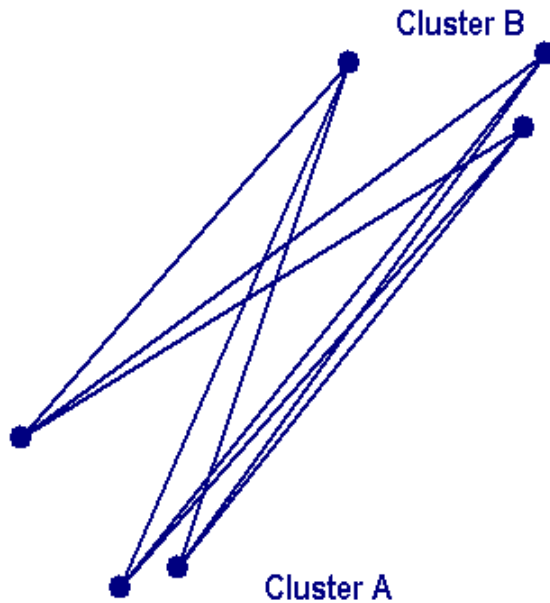


계층적 군집화: Hierarchical Clustering

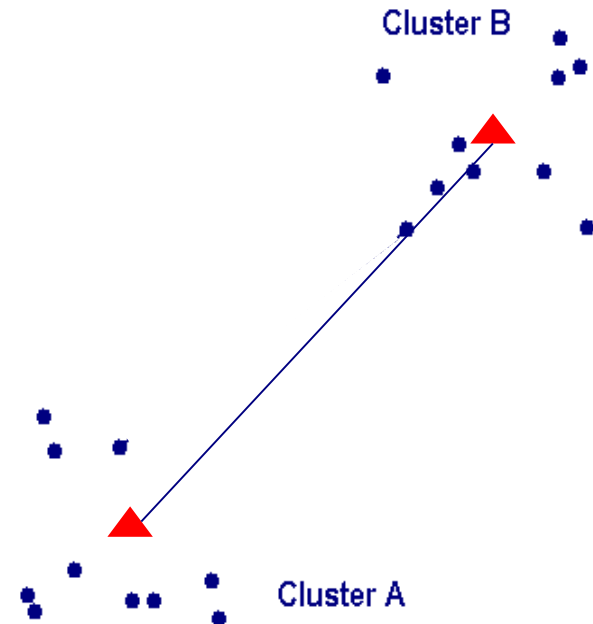
❖ 상향식 군집화 알고리즘

- Average linkage (mean distance): 각 군집에 속한 개체들 사이의 거리 평균값을 군집간 거리로 정의
- Centroid linkage (distance between centroids): 각 군집의 중심간 거리를 군집간 거리로 정의

Average linkage



Centroid linkage



계층적 군집화: Hierarchical Clustering

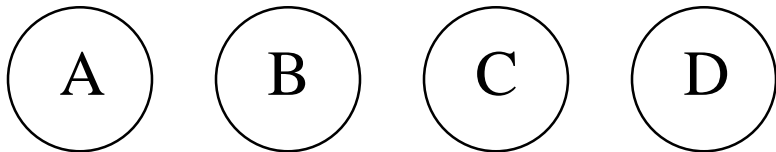
❖ 상향식 군집화 알고리즘

- 1단계: 모든 개체를 개별 군집으로 정의하고 군집간 거리 행렬 계산
- 2단계: 다음 절차를 반복
 - ✓ 2-1단계: 가장 가까운 두 개의 군집을 하나의 군집으로 통합
 - ✓ 2-2단계: 군집간 거리 행렬 업데이트
- 종료 조건: 모든 개체가 하나의 군집으로 통합되면 종료

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Initial Data Items



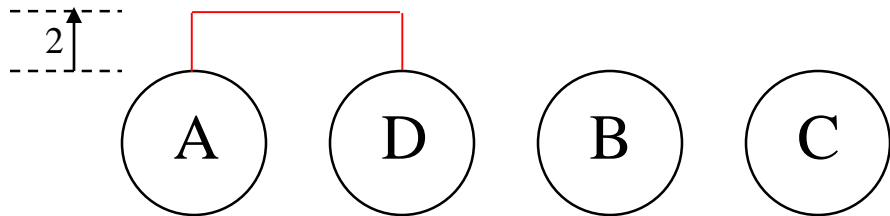
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



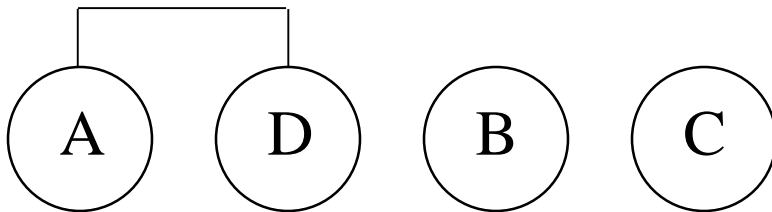
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



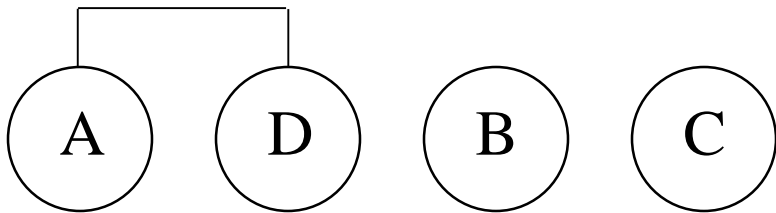
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



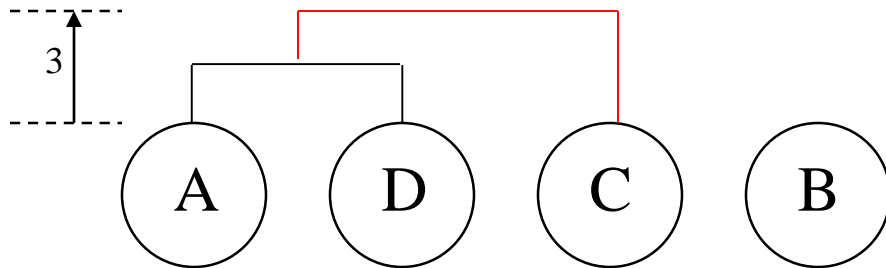
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



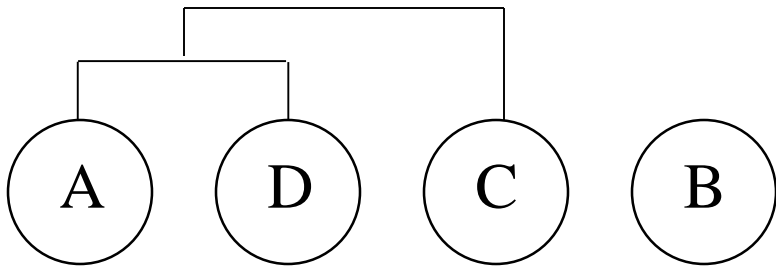
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



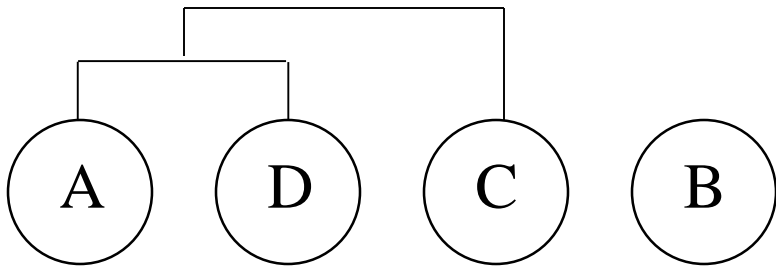
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



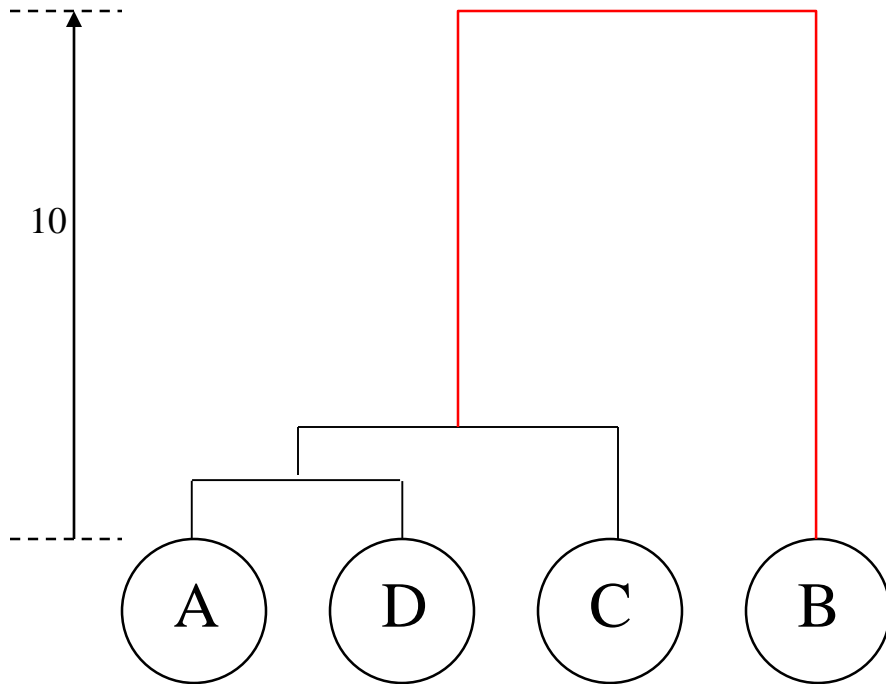
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



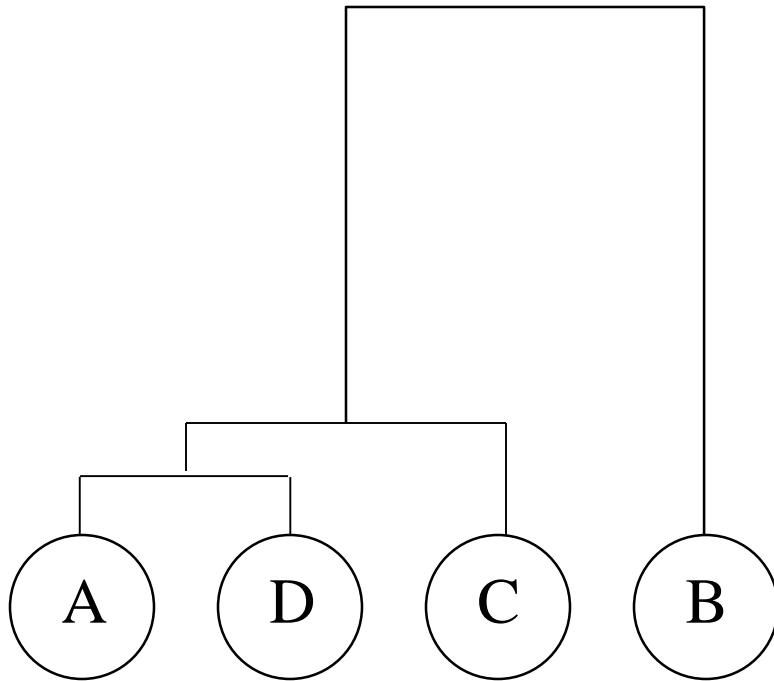
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Final Result



Distance Matrix

Dist	AD CB			
AD CB				

- 미혼 남성
- 미혼 여성
- 기혼 남성
- 기혼 여성
- 고령 기혼 여성

[illegible]

목차

I

군집화 소개

II

K-평균 군집화: K-Means Clustering

III

계층적 군집화: Hierarchical Clustering

IV

자기 조직화 지도

V

R 실습

자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기 조직화 지도: Self-Organizing Map (SOM)

- 고차원의 데이터를 사람이 시각적으로 이해할 수 있는 저차원(2차원 또는 3차원) 격자에 표현하는 방식
 - ✓ 고차원에서 유사한 개체들은 저차원에 인접한 격자들과 연결됨
 - ✓ 인공신경망 학습 알고리즘을 차용: 비지도적 경쟁학습
- 저차원의 격자에서의 유사도는 고차원 입력 공간에서의 유사도를 최대한 보존하도록 학습

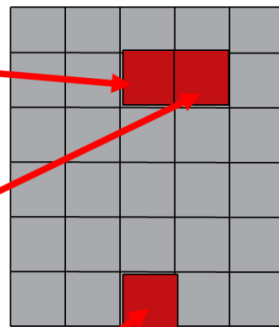
Input Pattern 1



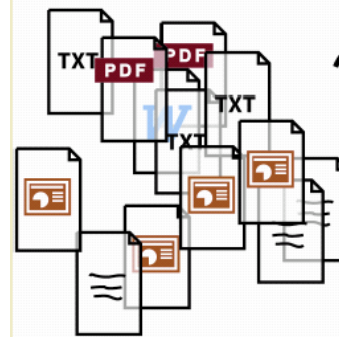
Input Pattern 2



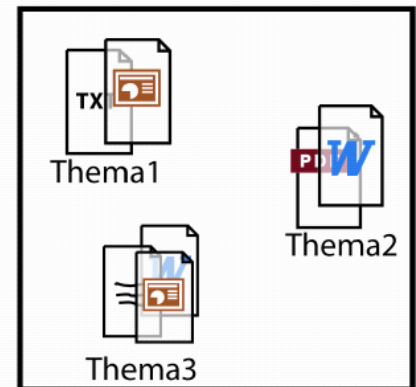
Input Pattern 3



Dokumenten- /
Informationssammlung



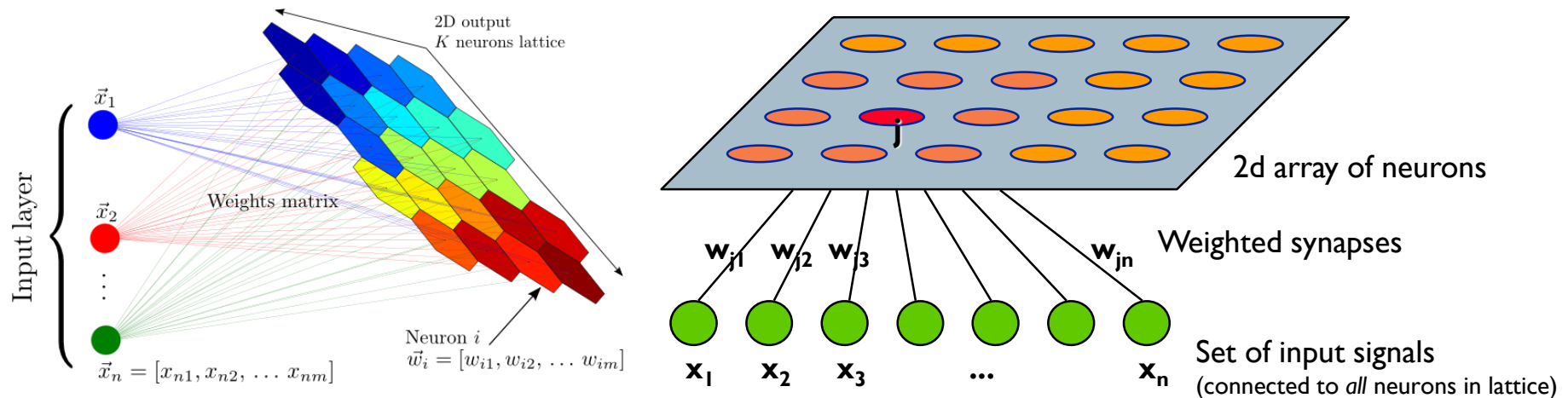
Semantische Karte



자기조직화 지도: Self-Organizing Map (SOM)

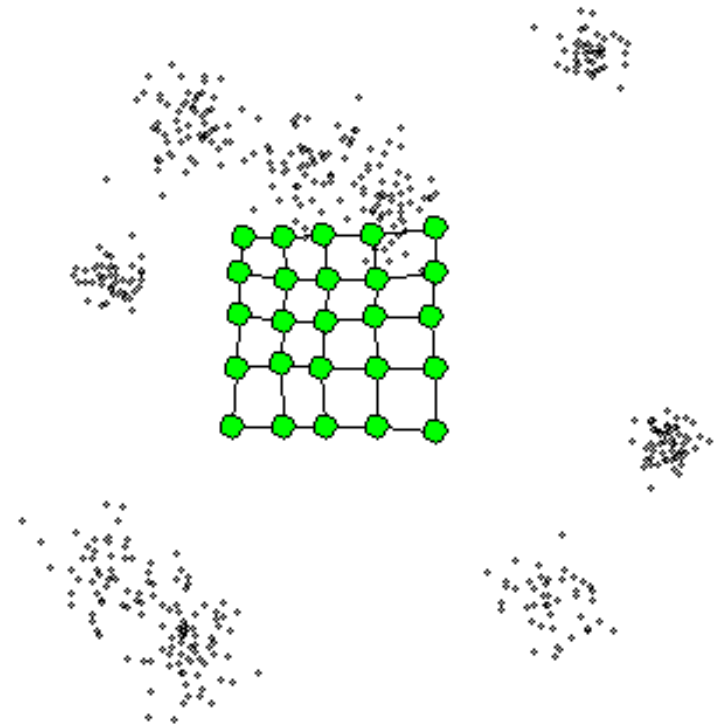
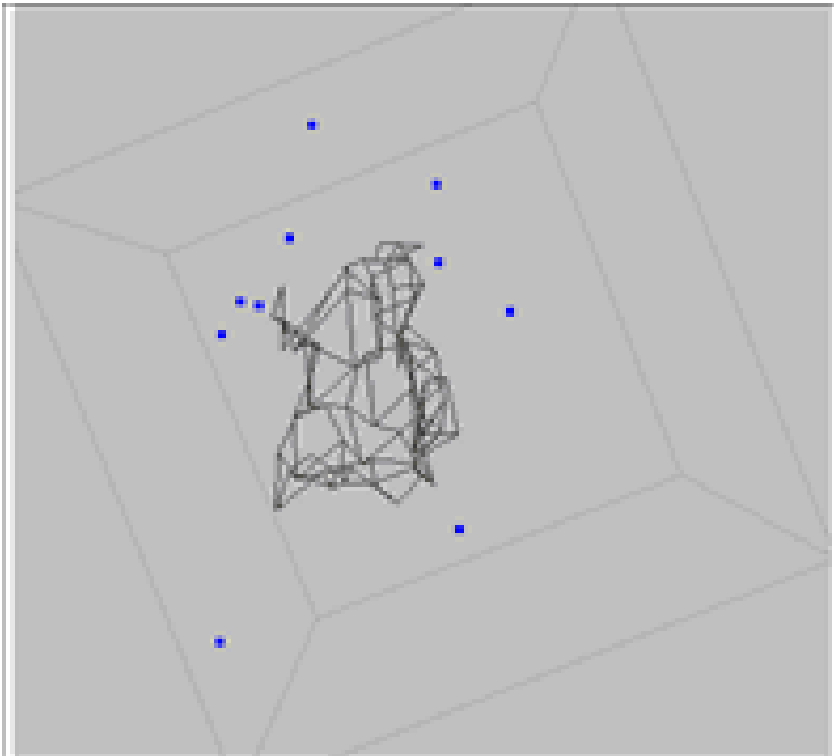
❖ 자기조직화 지도: 구조

- 저차원 격자의 모든 노드는 원 공간의 모든 개체들과 가중치 w 로 연결되어 있음
- 저차원 격자의 노드들은 서로 위치적인 유사도 관계를 가짐
- 원 공간의 각 개체와 가장 유사한 형태의 가중치를 갖는 **Winning 노드**가 선택됨
- 선택된 노드 및 근처 노드들이 **활성화**되어 원 공간의 개체와 유사하도록 가중치를 조정함



자기조직화 지도: Self-Organizing Map (SOM)

❖ SOM 예시



자기조직화 지도: Self-Organizing Map (SOM)

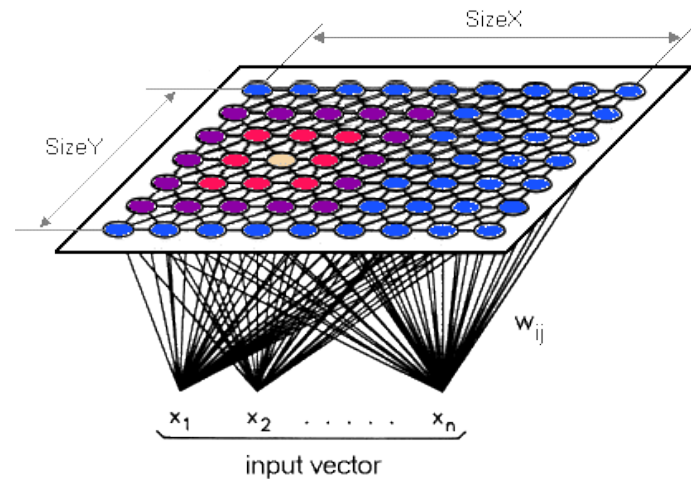
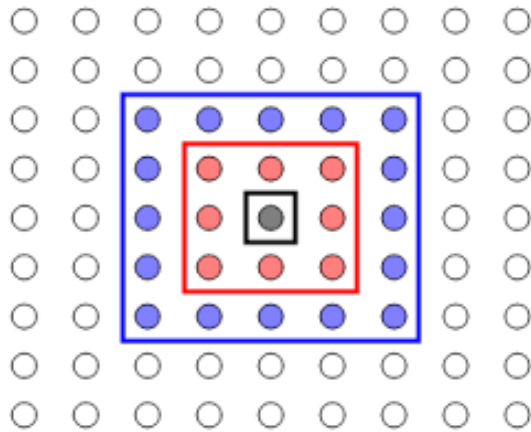
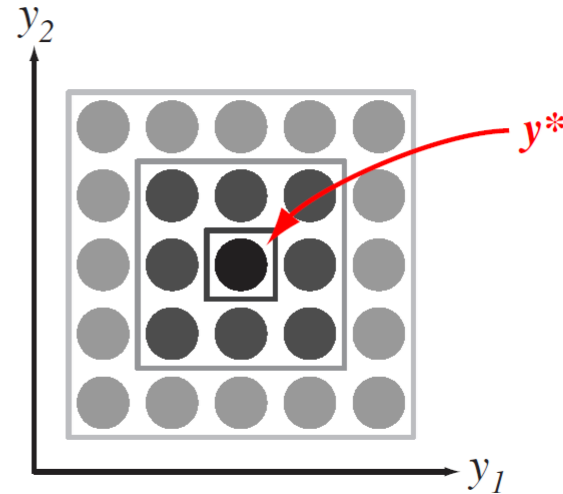
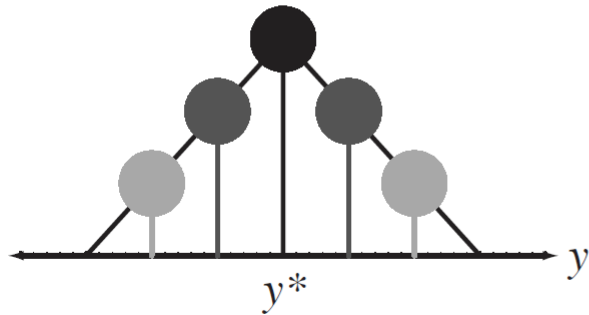
❖ 자기조직화지도: 학습

1. 자기조직화지도 격자 설정
2. 각 노드의 가중치를 설정 (usually at random)
3. 학습 데이터의 한 개체에 대해 모든 격자와의 유사도를 평가하여 Best Machine Unit (BMU) 선택
4. BMU와 이웃 노드와의 유사성 계산
5. BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, 이웃노드들도 일정 수준 가중치 업데이트를 수행
6. 가중치의 변화가 없을 때까지 Step 3부터 Step 5까지를 반복

자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화지도: 학습

4. 이웃노드와의 유사성 계산



자기조직화 지도: Self-Organizing Map (SOM)

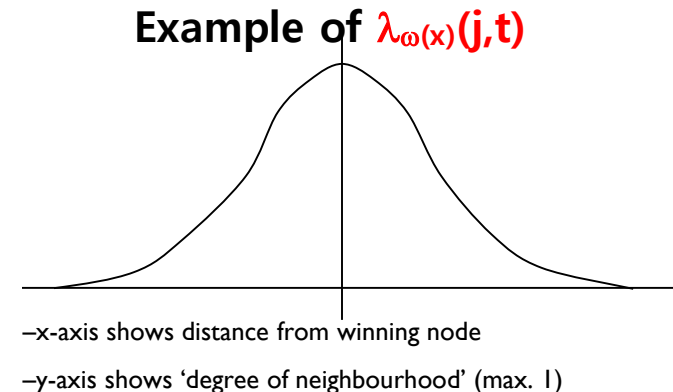
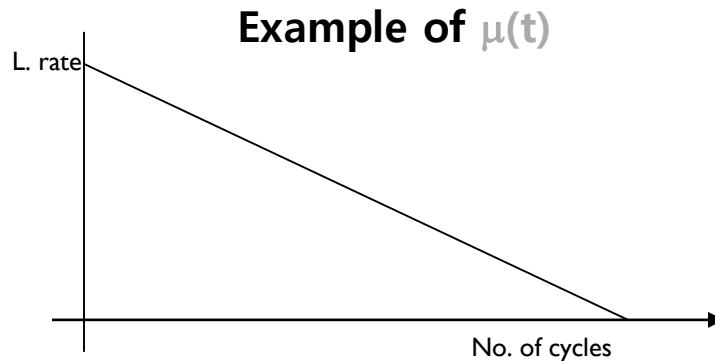
❖ 자기조직화지도: 학습

5. BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, 이웃노드들도 일정 수준 가중치 업데이트를 수행

<SOM Weight Update Equation>

$$w_j(t+1) = w_j(t) + \mu(t) \lambda_{\omega(x)}(j,t) [x - w_j(t)]$$

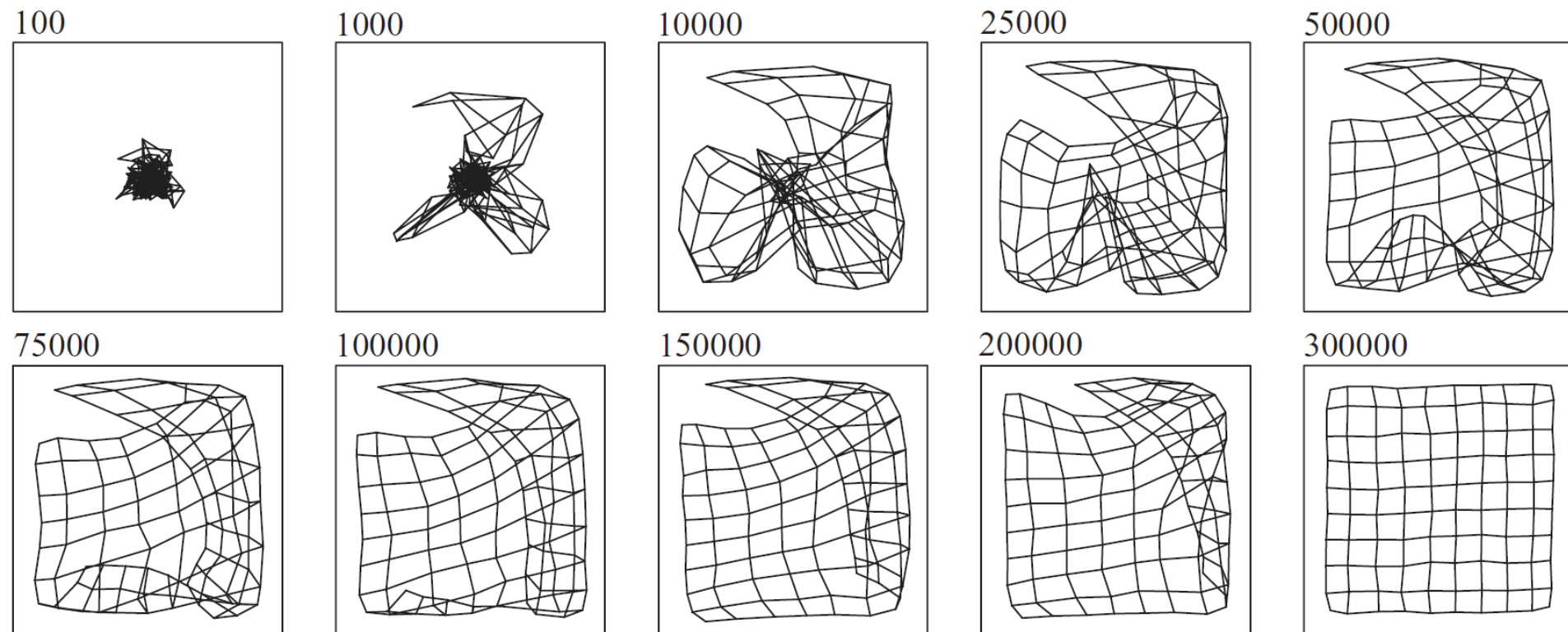
“The weights of every node are updated at each cycle by adding
 Current learning rate \times Degree of neighbourhood with respect to winner \times Difference
 between current weights and input vector
 to the current weights”



자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 수렴

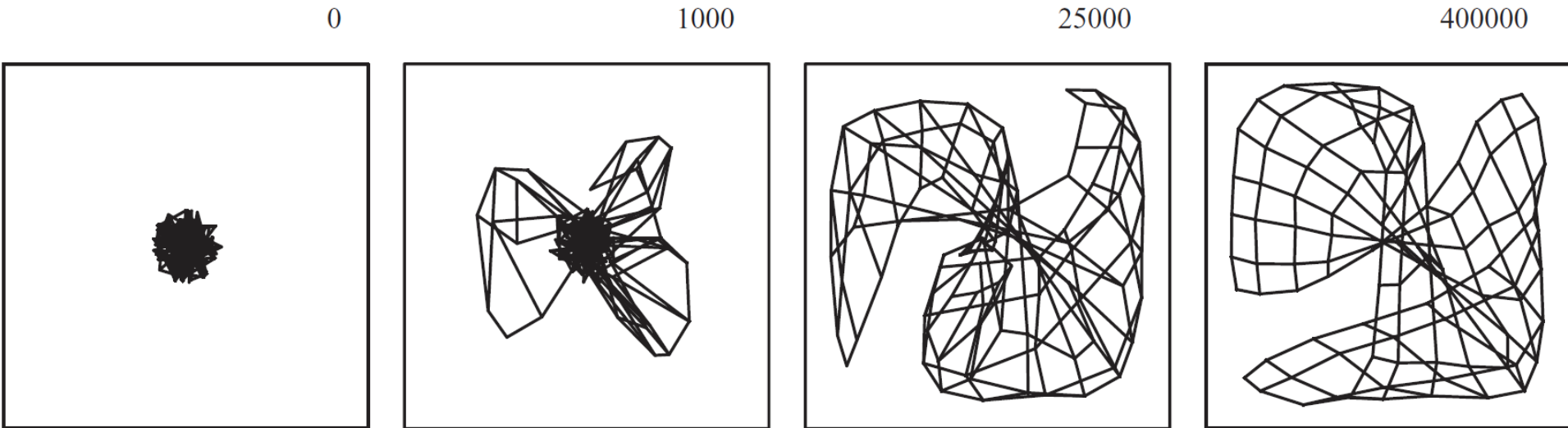
■ 행복하게도...



자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 수렴

■ 가끔씩은...



■ 초기 가중치를 재설정

목차

I

군집화 소개

II

K-평균 군집화: K-Means Clustering

III

계층적 군집화: Hierarchical Clustering

IV

자기 조직화 지도

V

R 실습

K-평균 군집화

❖ K-평균 군집화는 다양한 패키지에서 제공

- stats, kml, kml3d, RSKC, skmeans, sparcl, etc.

❖ R에서 기본 제공되는 Iris 데이터를 이용한 K-평균 군집화 수행

```
1 # Package for cluster validity
2 install.packages("clValid")
3 library(clValid)
4
5 # Load the Iris dataset
6 data(iris)
7
8 # Part 1: K-Means Clustering -----
9 # Remove the class label
10 newiris <- iris
11 newiris$Species <- NULL
12 rownames(newiris) <- paste("I", 1:150, sep = "_")
13
14 # Perform K-Means Clustering with K=3
15 kc <- kmeans(newiris,3)
16
17 str(kc)
18 kc$centers
19 kc$size
20 kc$cluster
```

- 각 군집의 중심, 군집에 속한 개체 수, 각 개체가 속한 군집 id 등

58/76

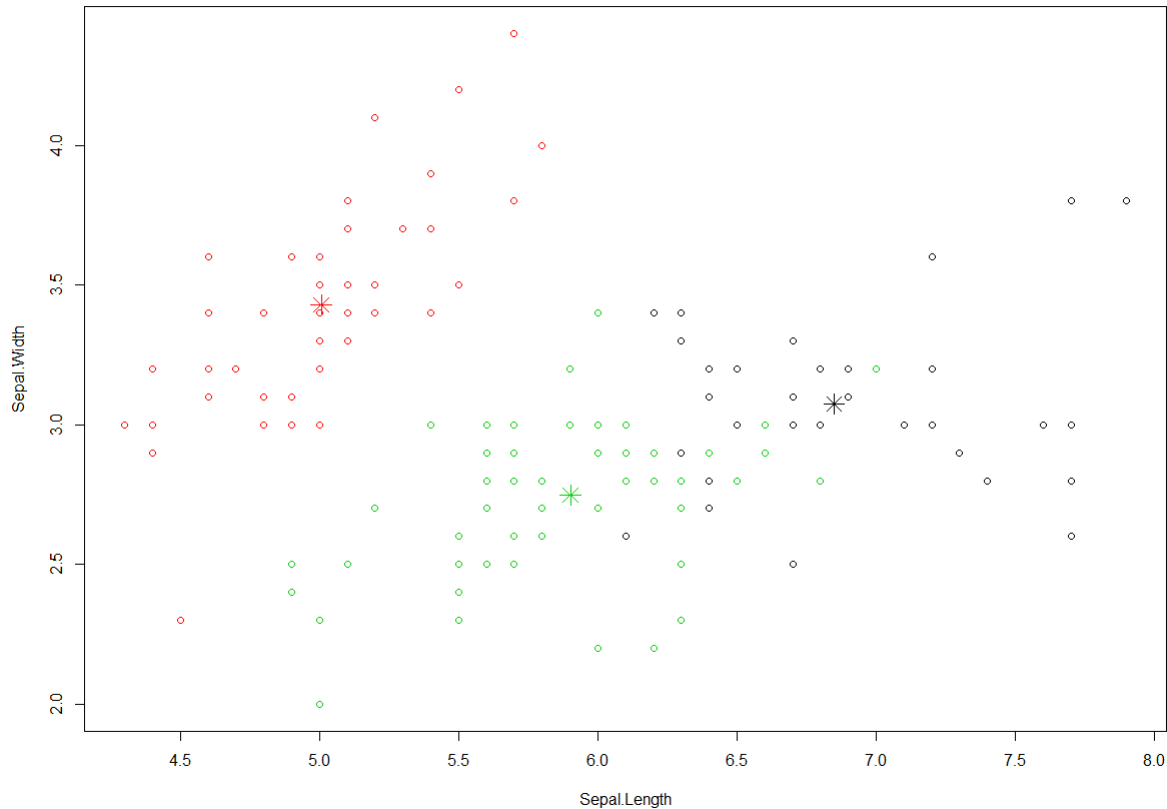
K-평균 군집화

❖ 군집화 결과 도시

```

25 plot(newiris[,c("Sepal.Length", "Sepal.Width")], col = kc$cluster)
26 points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex=2)

```



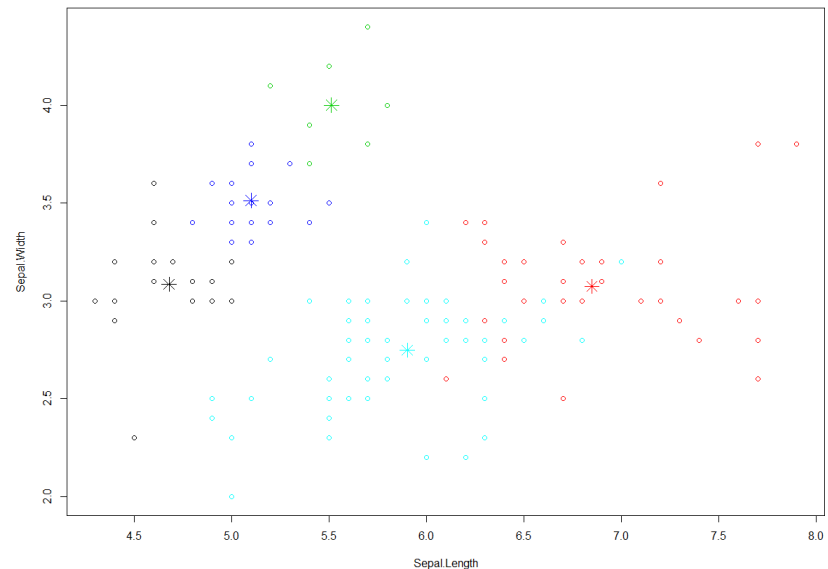
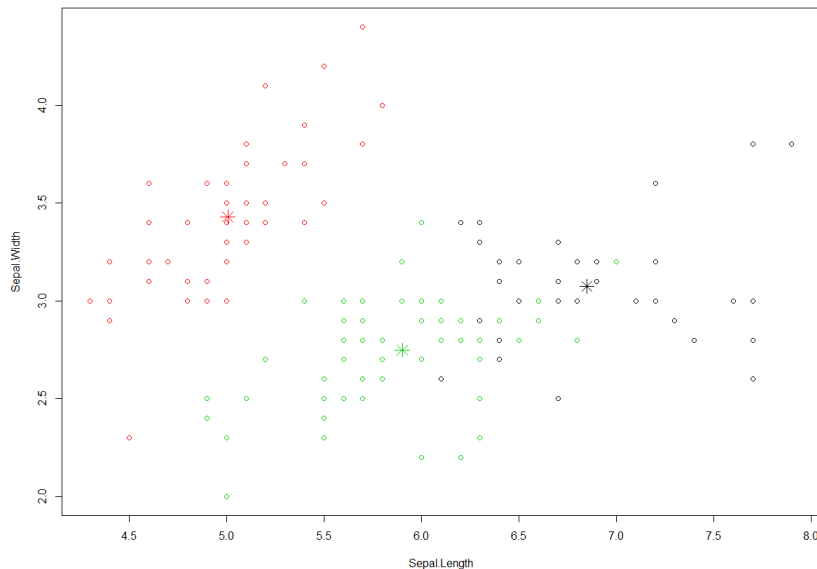
K-평균 군집화

❖ 군집의 수를 5로 다시 설정하고 K-평균 군집화 수행

```

28 # Perform K-Means Clustering with K=5
29 kc <- kmeans(newiris,5)
30
31 # Compare the assigned clusters and the Species
32 table(iris$Species, kc$cluster)
33
34 plot(newiris[,c("Sepal.Length", "Sepal.Width")], col = kc$cluster)
35 points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:5, pch = 8, cex=2)

```



K-평균 군집화

❖ 군집 타당성 지표 비교

```
37 # Evaluating the cluster validity measures
38 newiris.clValid <- clValid(newiris, 2:10, clMethods = "kmeans", validation = c("internal", "stability"))
39 summary(newiris.clValid)
```

```
> summary(newiris.clValid)
```

Clustering Methods:
kmeans

Cluster sizes:
2 3 4 5 6 7 8 9 10

Validation Measures:

		2	3	4	5	6	7	8	9	10
kmeans	APN	0.0130	0.0630	0.1572	0.2394	0.1680	0.1954	0.2212	0.2198	0.2619
	AD	1.2223	0.9390	0.8722	0.8149	0.7309	0.6946	0.6804	0.6489	0.6306
	ADM	0.0562	0.1131	0.2803	0.3316	0.2293	0.2340	0.2523	0.2245	0.2593
	FOM	0.4990	0.3935	0.3590	0.3534	0.3354	0.3144	0.3131	0.3050	0.3009
	Connectivity	6.1536	10.0917	17.5194	27.9373	36.4873	33.9595	38.9556	49.9901	58.0988
	Dunn	0.0765	0.0988	0.1365	0.0823	0.0853	0.0872	0.0872	0.0617	0.0684
	Silhouette	0.6810	0.5528	0.4981	0.4887	0.3648	0.3609	0.3556	0.3360	0.3391

Optimal Scores:

	Score	Method	Clusters
APN	0.0130	kmeans	2
AD	0.6306	kmeans	10
ADM	0.0562	kmeans	2
FOM	0.3009	kmeans	10
Connectivity	6.1536	kmeans	2
Dunn	0.1365	kmeans	4
Silhouette	0.6810	kmeans	2

K-평균 군집화

❖ Personal Loan 데이터를 이용하여 은행 고객 군집화

Data Description:

ID	Customer ID
Age	Customer's Age in completed years
Experience	#years of professional experience
Income	Annual income of the customer (\$000)
ZIPCode	Home Address ZIP code.
Family	Family size (dependents) of the customer
CCAvg	Avg. Spending on Credit Cards per month (\$000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. (\$000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a Securities account with the bank?
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by UniversalBank?

계층적 군집화

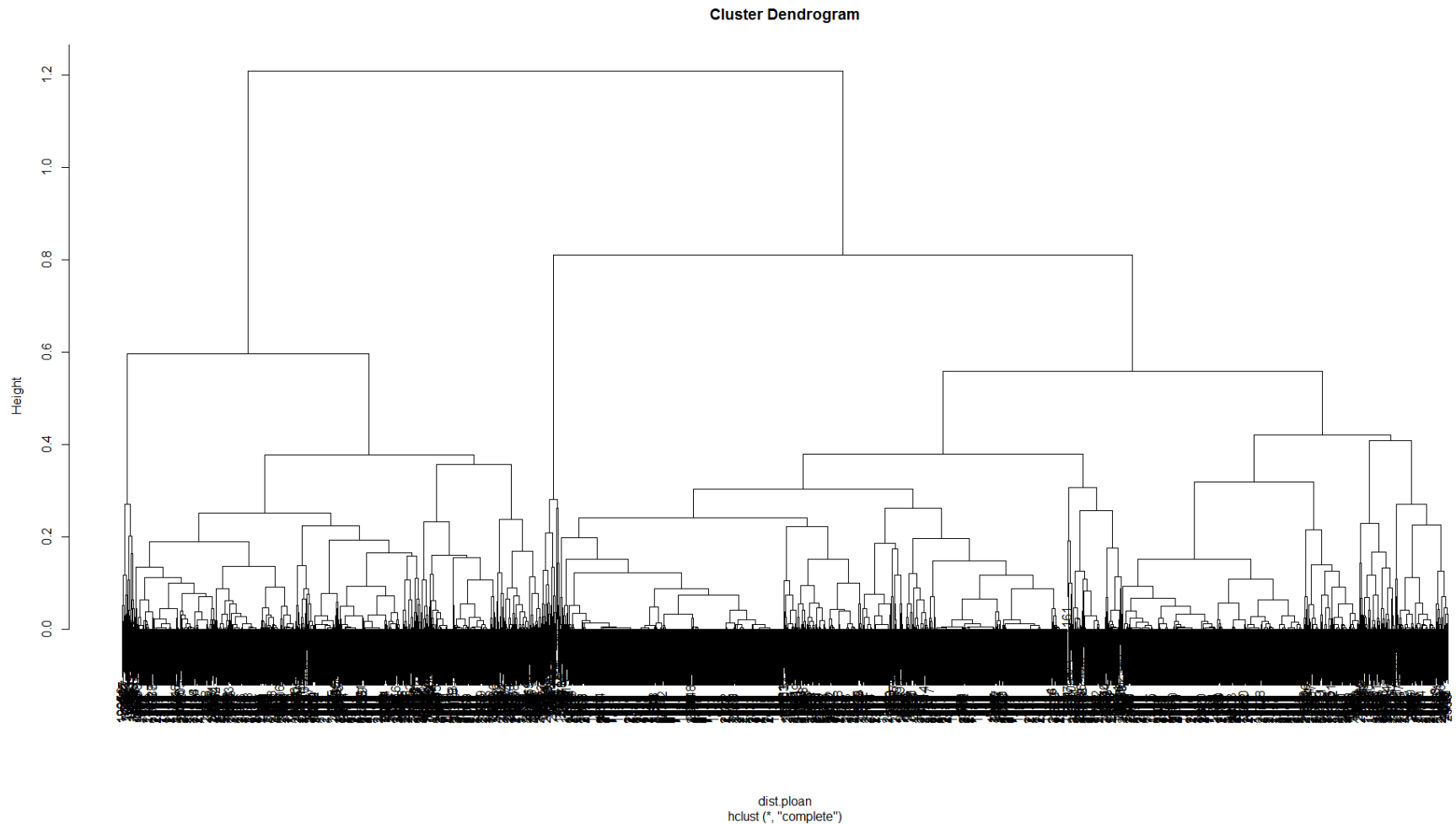
❖ Personal Loan 데이터를 이용하여 은행 고객 군집화

- 고객간의 유사도 측정은 피어슨 상관계수 사용
- 계층적 군집화를 위한 집단간 거리는 complete linkage 사용

```
41 # Part 2: Hierarchical Clustering -----  
42 ploan <- read.csv("Personal Loan.csv")  
43 ploan.x <- ploan[,-c(1,5,10)]  
44  
45 # Compute the similarity using the spearman coefficient  
46 cor.Mat <- cor(t(ploan.x), method = "spearman")  
47 dist.ploan <- as.dist(1-cor.Mat)  
48  
49 # Perform hierarchical clustering  
50 hr <- hclust(dist.ploan, method = "complete", members=NULL)
```

계층적 군집화

◆ 덴드로그램 생성



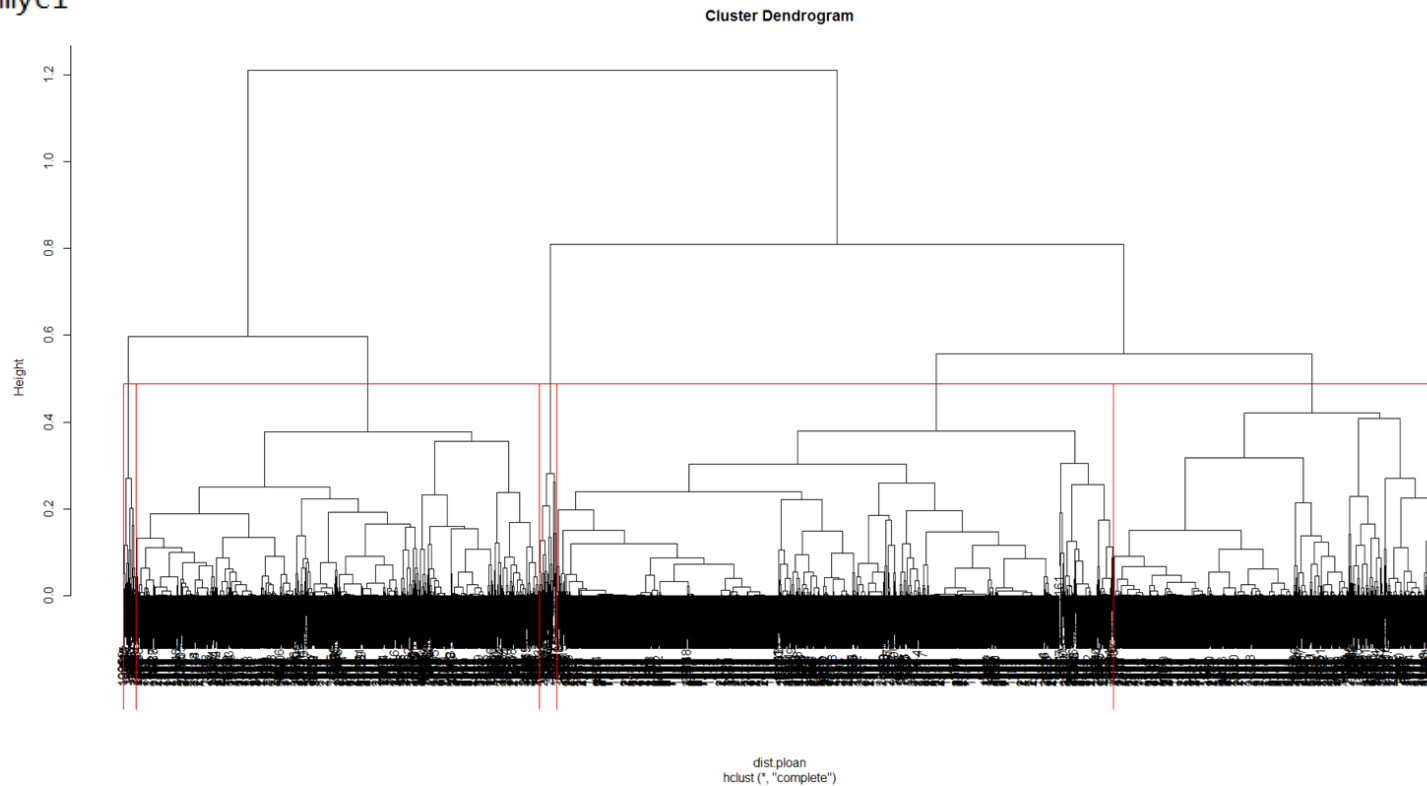
계층적 군집화

❖ 군집화 수행

```

52 # plot the results
53 plot(hr)
54 plot(hr, hang = -1)
55 plot(as.dendrogram(hr), edgePar=list(col=3, lwd=4), horiz=T)
56
57 # Find the clusters
58 mycl <- cutree(hr, k=5)
59 mycl

```



계층적 군집화

❖ 각 군집의 특성 분석

```

64 # Compare each cluster
65 segment.ploan <- cbind(ploan.x, ploanYN = ploan[,10], clusterID = as.factor(mycl))
66 segment.summary <- data.frame()
67
68 for (i in 1:(dim(segment.ploan)[2]-1)){
69   segment.summary = rbind(segment.summary,
70                           tapply(segment.ploan[,i], segment.ploan$clusterID, mean))
71 }
72
73 colnames(segment.summary) <- paste("cluster", c(1:5))
74 rownames(segment.summary) <- c(colnames(ploan.x), "LoanRatio")
75 segment.summary

```

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
Age	45.14919736	45.86962190	46.91558442	24.96969697	25.8400
Experience	19.96411709	20.67796610	21.62012987	-0.60606061	0.0800
Income	87.71671388	73.39504563	52.86688312	71.30303030	80.5200
Family	2.33899906	2.42503259	2.44805195	2.90909091	3.1600
CCAvg	2.69259679	1.86147327	0.78404221	1.82878788	2.2272
Education	1.71671388	1.86440678	2.09740260	2.18181818	2.0800
Mortgage	0.00000000	180.92568449	0.00000000	0.00000000	188.0400
Securities.Account	0.10103872	0.11473272	0.11688312	0.12121212	0.1200
CD.Account	0.06421152	0.07953064	0.04220779	0.03030303	0.0000
Online	0.46553352	0.59061278	0.83441558	0.57575758	0.6000
CreditCard	0.28706327	0.27770535	0.31168831	0.33333333	0.2400
LoanRatio	0.13408876	0.11734029	0.03246753	0.09090909	0.0400

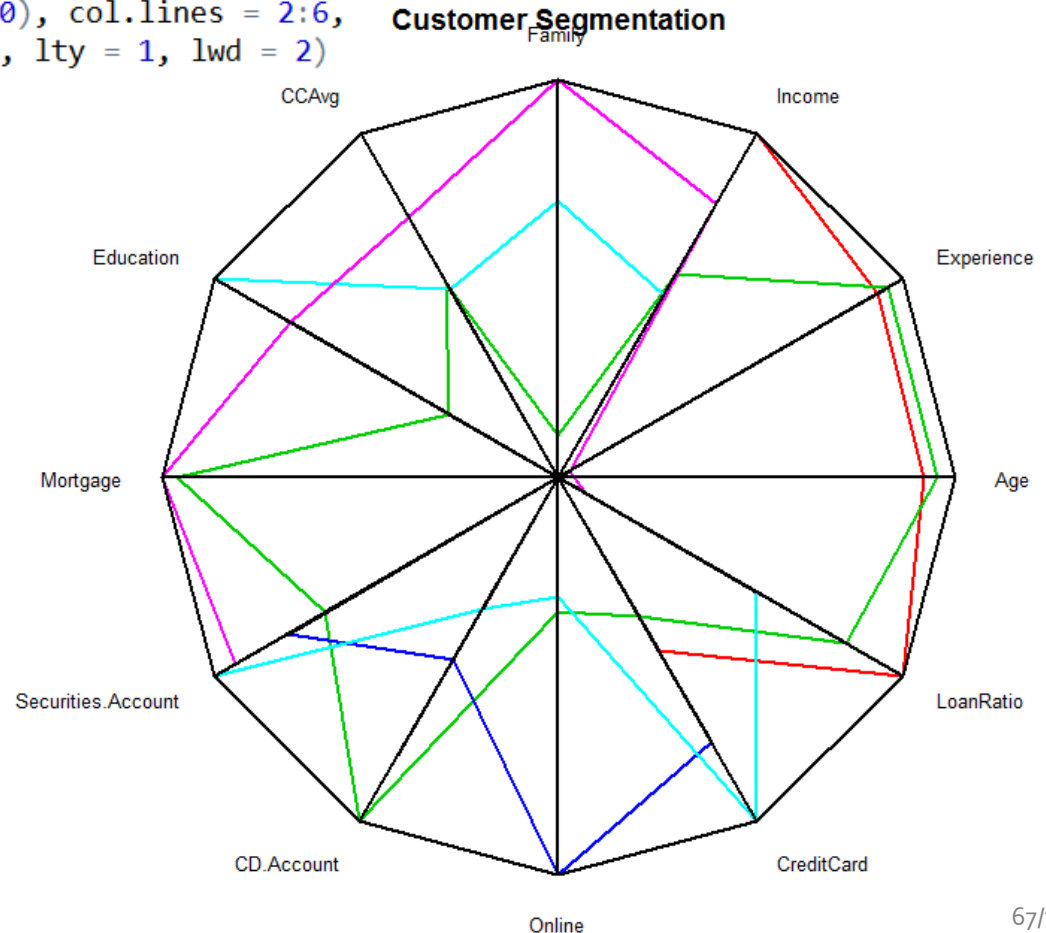
계층적 군집화

❖ 각 군집의 특성 분석

```

77 # Radar chart
78 segment.summary <- t(segment.summary)
79 stars(segment.summary, locations = c(0, 0),
80       radius = TRUE, key.loc = c(0, 0), col.lines = 2:6,
81       main = "Customer Segmentation", lty = 1, lwd = 2)
82

```



자기조직화지도: SOM

❖ SOM을 수행할 수 있는 패키지

- kohonen, som, wccsom, etc.

❖ som/kohonen packages 사용

❖ 대상 데이터: Yeast & Wine

- Yeast: 6601 genes measured at 18 time points
- Wines: 177 records with 13 characteristics

자기조직화지도: SOM

❖ Package som

```
83 # Part 3: Self-Organizing Map -----  
84 # som package install  
85 install.packages("som", dependencies = TRUE)  
86 detach("package:kohonen", unload=TRUE)  
87 library(som)  
88  
89 # Load the yeast dataset  
90 data(yeast)  
91 yeast <- yeast[, -c(1, 11)]  
92 yeast <- normalize(yeast, byrow=FALSE)  
93  
94 # Train SOM with two different settings  
95 som1 <- som(yeast, xdim=5, ydim=5, topol="rect", neigh="gaussian")  
96 som2 <- som(yeast, xdim=5, ydim=5, topol="hexa", neigh="bubble")  
97  
98 # See the results  
99 summary(som1)  
100 plot(som1)  
101 som1$visual[1:10,]  
102  
103 summary(som2)  
104 plot(som2)  
105 som2$visual[1:10,]
```

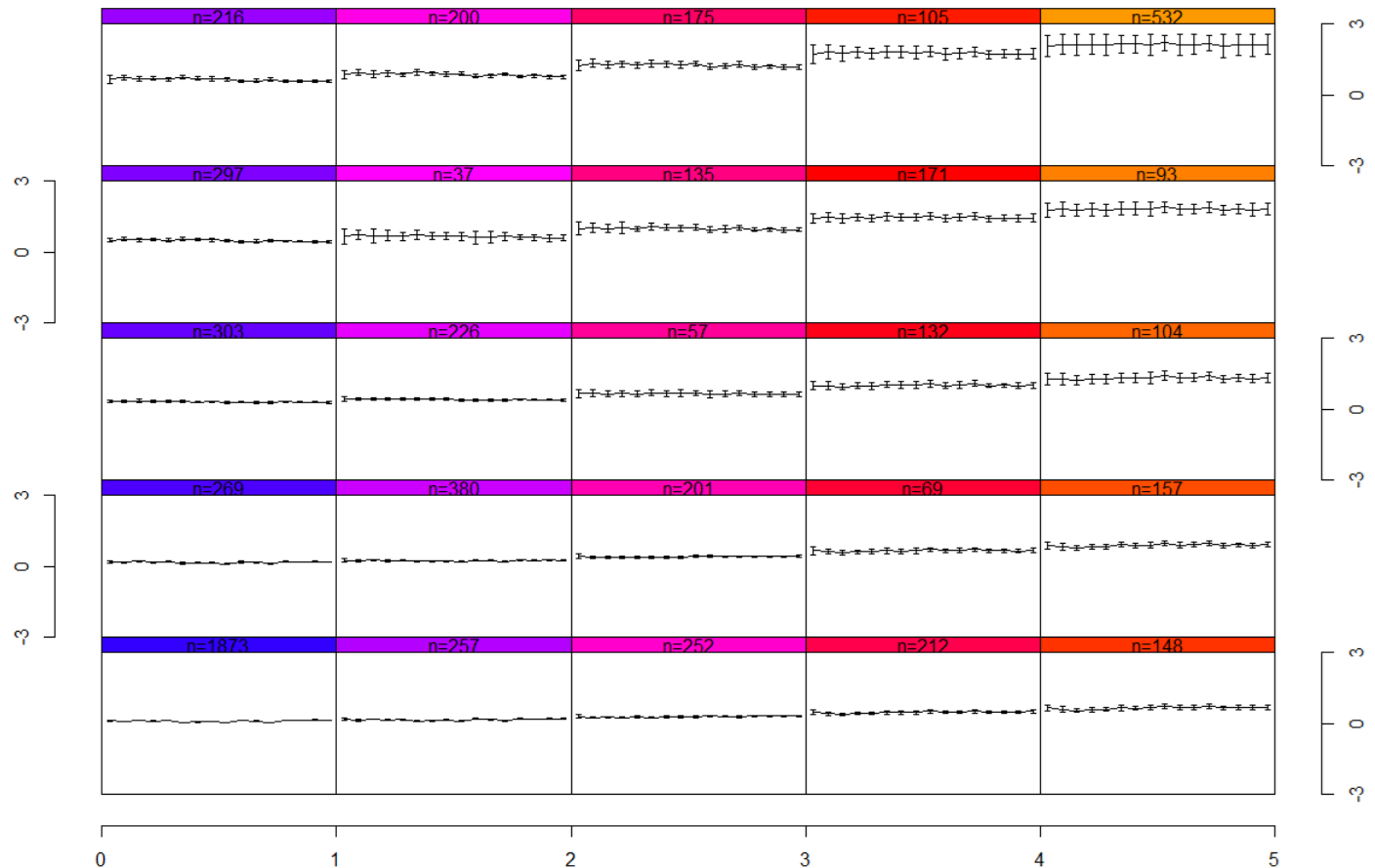
자기조직화지도: SOM

❖ Package som

```
> summary(som1)
Initialization: linear
Topology: rect
Neighborhood type: gaussian
Learning rate type: inverse
Initial learning rate parameter: 0.05 0.02
Initial radius of training area: 5 3
Average quantization error: 0.9505139
Average distortion measure: 25.37834 with error radius: 1
> som1$visual[1:10,]
      x y    qerror
1  0 0 0.6258783
2  0 0 0.8317910
3  0 0 0.7678554
4  0 0 0.7578742
5  0 0 0.7031048
6  0 0 0.6870237
7  0 0 0.7565792
8  0 0 0.6441827
9  0 0 0.7953218
10 0 0 0.6724060
```

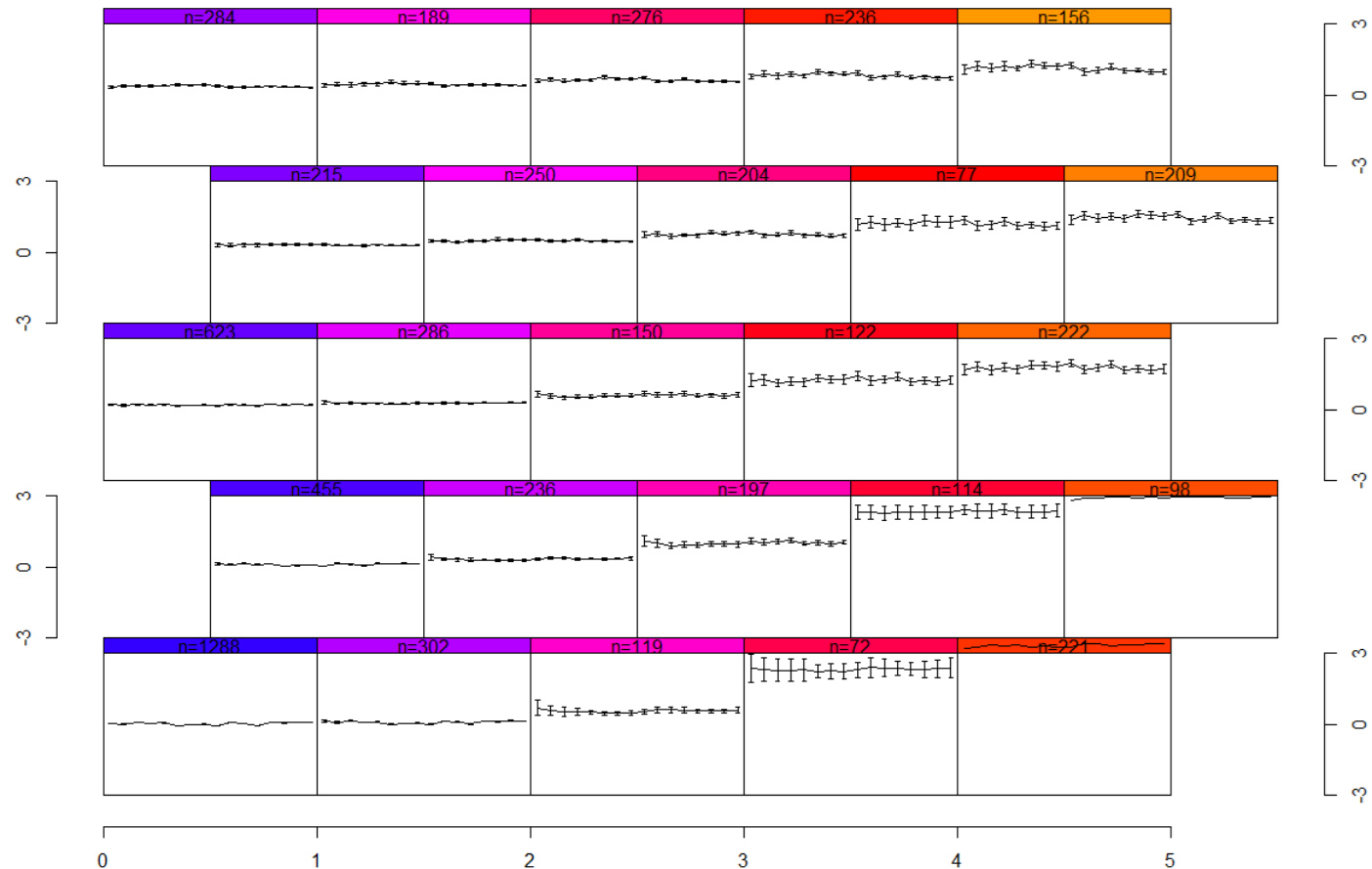
자기조직화지도: SOM

❖ Package som



자기조직화지도: SOM

❖ Package som



자기조직화지도: SOM

❖ Package kohonen

```

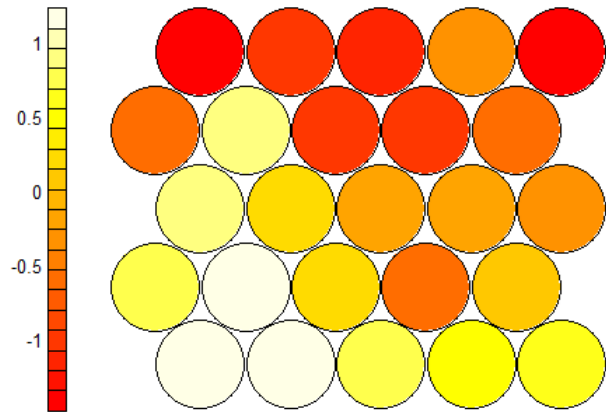
107 # kohonen package install
108 install.packages("kohonen", dependencies = TRUE)
109 detach("package:som", unload=TRUE)
110 library(kohonen)
111
112 data(wines)
113 trn = sample(nrow(wines), 120)
114 wines_trn <- scale(wines[trn,])
115 wines_tst <- scale(wines[-trn,], center = attr(wines_trn, "scaled:center"), scale = attr(wines_trn, "scaled:scale"))
116
117 som.wines <- som(wines_trn, grid = somgrid(5,5,"hexagonal"))
118 map(som.wines, wines_tst)
119 wine.cluster <- cutree(hclust(dist(som.wines$codes)), 3)
120
121 graphics.off()
122 par(mfrow = c(2, 2))
123 plot(som.wines, type = "property", property = som.wines$codes[,1], main = colnames(som.wines$codes)[1])
124 plot(som.wines, type = "property", property = som.wines$codes[,2], main = colnames(som.wines$codes)[2])
125 plot(som.wines, type = "property", property = som.wines$codes[,3], main = colnames(som.wines$codes)[3])
126 plot(som.wines, type = "property", property = som.wines$codes[,4], main = colnames(som.wines$codes)[4])
127 graphics.off()
128
129 par(mfrow = c(2, 2))
130 plot(som.wines, type = "quality")
131 plot(som.wines, type = "codes")
132 plot(som.wines, type = "changes")
133 dev.off()
134
135 som.pal <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b", "#e377c2")
136 plot(som.wines, type="mapping", bgcol = som.pal[wine.cluster], main = "Clusters")
137 add.cluster.boundaries(som.wines, wine.cluster)

```

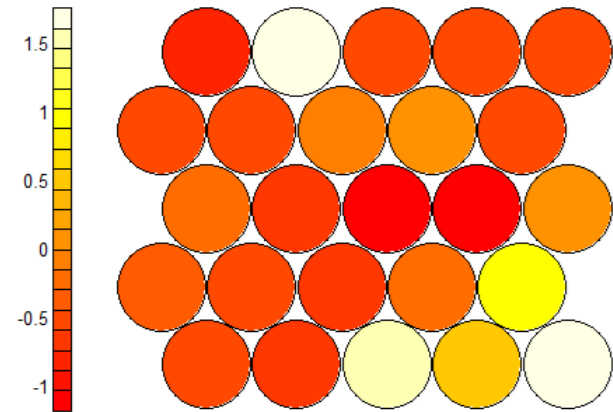
자기조직화지도: SOM

❖ Package kohonen

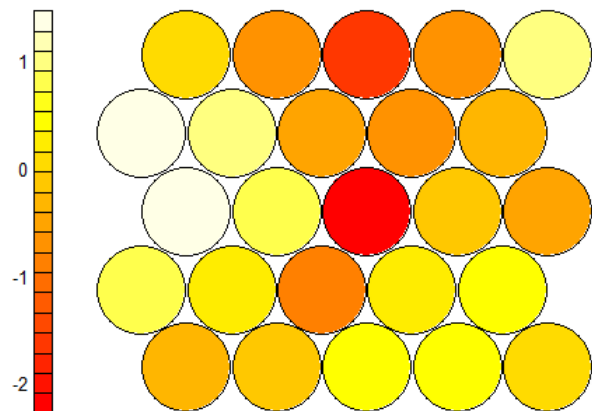
alcohol



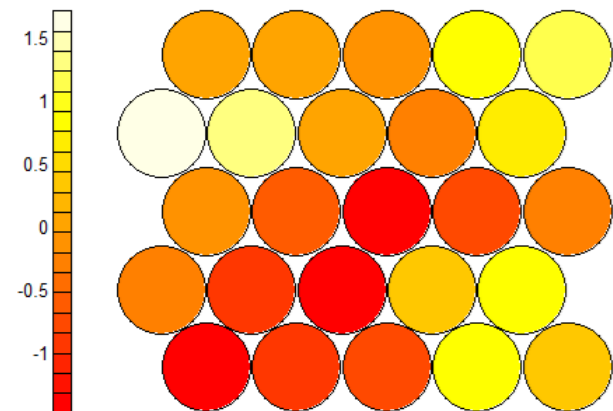
malic acid



ash



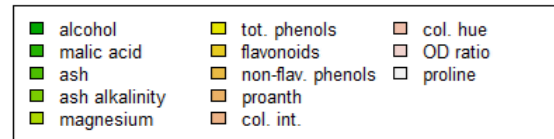
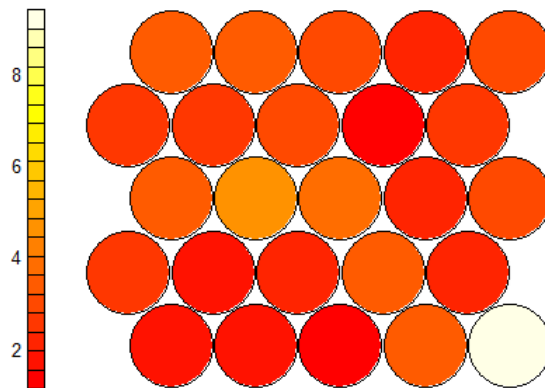
ash alkalinity



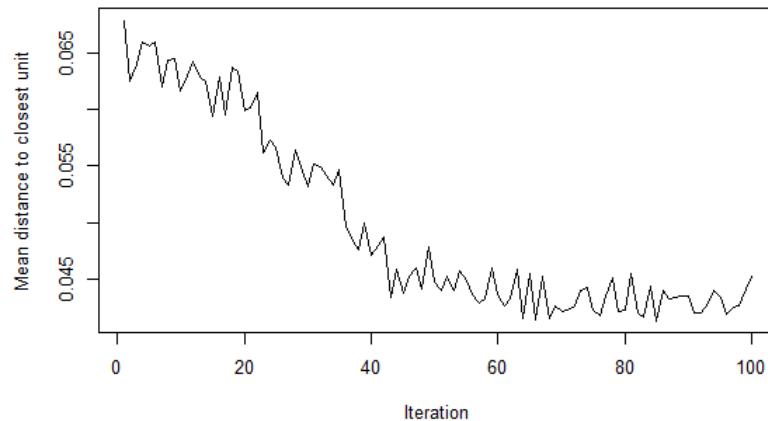
자기조직화지도: SOM

❖ Package kohonen

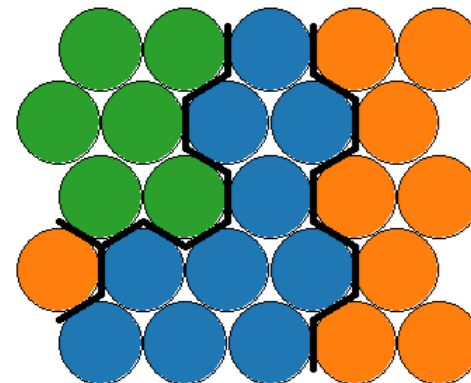
Distance plot



Training progress



Clusters



Q & A

